U-1

Youtube : Point of Programming

Introduction

Data:-

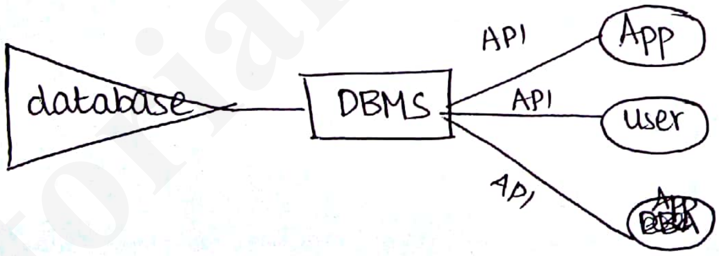
Data is a collection of facts and figures that can be processed to produce information.

Data Base:-

Data Base is a collection of related data. That related data is organised so that it can be easily accessed, managed and updated.

Data Base Management Systems:-

Database management system is the software that allows a computer to perform database functions of storing, retreiving, adding, deleting and modifying data.



Main objectives of DBMS:-

1. Provide for mass storage of relavant data.
2. Making easy access to data for the authorized user.
3. providing prompt response to users requests for data
4. Eliminate duplicate data.

5. Allow multiple users to be active at one time.

6. Allow the growth of database system.

7. provide data integrity.

8. protect the data from physical harm and unauthorized access.

9. serving different types of users.

10. provide security with user access privilege.

11. combining interrelated data to generate report.

12. provide multiple views for same data.

Content: File System v/s DBMS

1.

Frequently Asked questions:-

1. List the disadvantages of file processing model - 4M
2. List any four features of that a data base system provides two uses - 3M.
3. Explain in detail about DBMS advantages over file management system. - 8M
4. compare DBMS system with conventional file system - 8M
5. What are the main characteristics of DBMS and how it differ from file system - 8M.
6. Difference between file system vs DBMS - 8M

ANSWER:-

S.No	Difference factor	File System	DBMS
1.	Definition	A file management system is an abstraction to store, retrieve, management and update a set of files. A file management system keep track on the files and also manage them	Database Management system (DBMS) is a collection of interrelated data and a set of programs to access those data. Some of the very well known DBMS are microsoft Access, Microsoft SQL Server, oracle, SAP, Fox pro etc.

2.	Data Redundancy	In file system approach, Each user defines and implements The needed files for a specific application to run. For Ex, in sales department of an Enterprise, one user will be maintaining The details of how many sales personnel are there in The sales department and their grades. Another user will be maintaining The sales person salary details	Although The database approach doesnot remove redundancy completely, it controls The amount of redundancy in The database because in database approach, a single repository of data is maintained That is defined once and then accessed by many users. The fundamental characteristic of database approach is that The database system not only contains data's but it contains complete definition of The database structure and constraints.

3.	sharing of data	file system doesn't allow sharing of data	In DBMS data can be shared very Easily due to centralized system.
4	Data consistency	When data is redundant, it is difficult to update. for Ex, if we want to change (or) update Employee's address, then we have to make changes at all the places where data of that employee is stored. if by mistake, we forgot to change or update The address at one or more places then data inconsistency will occur. i.e, The appearance of same data will differ from Each other	in DBMS, as there is no or less data redundancy, data remains consistent.
5.	Difficult to search/access data	In conventional file system, if we want to search/retrieve/access some data item, it becomes very difficult because in file system for every	In DBMS searching / retrieval/ accessing of data item is very easy and user friendly because searching & querying

		operation we have to make different programs	operations are already available in The system.
6	Data isolation	In file system there is no standard format of data or we can say data is scattered in various formats or files which also make data retrieval difficult	In DBMS, due to centralized system The format of similar type of data remains same.
7	Data Integrity	The Value of data in database must follow or satisfy some rules or consistency constraints. for Ex, A company have a policy that the age of an Employee must be >=18. The Value which is not satisfying this constraint must not be stored in The respective column. In file system, there is no procedure to check	DBMS maintains The data integrity by enforcing the constraints by adding appropriate code.

8.	Security Problems	In file system there is no or very less security. General security provided by file system are locks, guards etc	DBMS have high level security like encryption, passwords, biometric security etc.
9.	Atomicity	Atomicity means a transaction must be all-or-nothing i.e, the transaction must either fully happen, or not happen at all. It must not complete partially. Eg. if A want to transfer 5000rs to B's a/c. In this case A's a/c should be debited and B's a/c should be credited with the same amount. let suppose A's a/c is debited with 5000rs and then transaction fails. Now the	Transaction atomicity is a special feature of DBMS. In DBMS either a transaction completed fully or none of the action is performed. For this, DBMS maintains the transaction log in which intermediate values are stored.

		transanction is incomplete because B's a/c is not credited. These type of problems occur in file system because there is no Procedure to stop such type of anomalies.	{Transanction atomicity is a special feature of DBMS. In DBMS either a transanction completed fully or none of the action is performed. For this, DBMS maintains the transanction log No in which intermediate values are stored}
10.	concurrent Access Anomalies	Any multi-user database application has to have some method for dealing with concurrent access to data -- when more than one user is accessing the same data at the same time. A problem occurs when user X reads a row for editing, user Y reads the same row for editing, user X saves changes The user X saves changes. The changes	

		made by user Y are lost unless something prevents user x from blindly overwriting the row. -file system does not provide any procedure to stop such type of anomalies.	DBMS along with an appropriate application provides safety towards concurrent access. for this locks are available in DBMS. If 2 or more transactions want to change/update or write a data item, an exclusive lock is issued to one of these transactions. until and unless the transaction release that lock no other transaction can acquire the lock & hence can't update/write the data item

②

Concept : DataBase Users (Actors on scene, Workers behind the scene).

FAQ's (4m and 8m)

1) What are the activities of database users?
2) List different types of database users
3) Explain various types of users and explain about their role in detail. (8m)
4) Responsibilities of DBA? (3m)
5) What is DBA? Explain the functions of DBA? (4m)
6) Role of DBA? (8M).

Answer.

Introduction: These apply to "large" databases, not "personal" databases that are defined, constructed and used by single person via say Microsoft Access.

* Users may be divided into:

→ Those who actually use and control database content and those who design, develop and maintain database application are called "Actors on the Scene".

→ Those who design and develop the DBMS software and related tools and computer systems operators (called "Workers behind the scene".

Actors on scene :

1. DataBase Administrator (DBA)
2. Database designers
3. End users
    - → Casual end users
    - → Naive / Parametric end users
    - → Sophisticated end users
    - → Stand-alone users
4. System analysts, Application Programmers, Software Engineers.

① DATABASE ADMINISTRATOR (DBA)

This is the chief administrator, who oversees and manages the database system (including data & software). He performs no. of crucial tasks. He must be on call all time when a problem does arise.

Duties / Roles / Functions / Responsibilities of DBA

1. Generate database application performance reports
2. Investigate user performance complaints
3. Modify database structure such as tables, indexes.
4. Evaluate and implement new database management features

5. Tune Database

6. Database planning

7. Policies and procedures

8. Logical/Conceptual database design

9. Security, Integrity & control

10. Maintain OS

11. Enrolling new users

12. Helping Load/unload data

13. Make data easily ACCESSIBLE when required.

14. Planning future storage requirements.

15. Data Repository (is a logical partitioning of data where multiple databases which apply to specific applications)

16. Data Replication (Process of storing data in more than one site or node. This is necessary for improving availability of data).

17. Creating primary objects once application developers have designed an application.

18. Creating primary database storage structures (table spaces) once application developers have designed an application.

19. Installing, upgrading, configure the oracle database

server and application tools

20. Primary role is to protect files from loss or corruption.

21. Maintains backups and recovery.

→ Backups can be used to restore & recover data that has been lost through server failure but different errors may need different recovery strategies. So, DBA must be alert to the dangers.

* DBA is a single person in organization with final responsibility and authority for database setup, use and maintenance.

② DATABASE DESIGNERS

* They are responsible for identifying data to be stored and for choosing an appropriate way to organize it.

* They also define views for different categories of users.

* Final design must be able to support the requirements of all user subgroups.

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- **Algorithms Notes**
- **Artificial Intelligence**
- **Android Programming**
- **C & C++ Programming**
- **Combinatorial Optimization**
- **Computer Graphics**
- **Computer Networks**
- **Computer System Architecture**
- **DBMS & SQL Notes**
- **Data Analysis & Visualization**
- **Data Mining**
- **Data Science**
- **Data Structures**
- **Deep Learning**
- **Digital Image Processing**
- **Discrete Mathematics**
- **Information Security**
- **Internet Technologies**
- **Java Programming**
- **JavaScript & jQuery**
- **Machine Learning**
- **Microprocessor**
- **Operating System**
- **Operational Research**
- **PHP Notes**
- **Python Programming**
- **R Programming**
- **Software Engineering**
- **System Programming**
- **Theory of Computation**
- **Unix Network Programming**
- **Web Design & Development**

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

③ **END USERS**: These are the persons who access the database for querying, updating and report generation.

* They are main reason for database's existence

→ Some of the users are:

(i) **Casual end users**:

* These use database occasionally, needing different information each time; use query language to specify their requests; typically middle or high level managers

(ii) **Naive/Parametric end users**:

* Typically biggest group of users; frequently query/update the database using standard canned transactions that have been carefully programmed and tested in advance.

Examples:

- Bank tellers check account balances, post deposits/withdrawals.
- Reservation clerks for airlines, hotels etc checks availability of seats/rooms & make reservations.
- Shipping clerks (eg:.at UPS) use buttons, bar code scanners etc to update status of packages.

(iii) Sophisticated end users: Engineers, scientists, business analysts who implement their own applications to meet their complex needs.

(iv) Standalone users: Use "personal" databases, possibly employing a special purpose (ex: financial).

* Mostly maintain personal databases using ready-to-use packaged applications

④ SYSTEM ANALYSTS, APPLICATION PROGRAMMERS, SOFTWARE ENGINEERS:

* System analysts: Determine needs of end users; especially naive and parametric users and develop specifications for canned transactions that meet these needs

* Application programmers: Implement, test, document, and maintain programs that satisfy the specifications.

* SOFTWARE ENGINEERS: Used to write the programs in order to access database efficiently.

WORKERS BEHIND THE SCENE

1. DBMS system designers/implementors
2. Tool developers
3. Operators & maintenance personel

① DBMS system designers/implementors:

Provide DBMS software that is at foundation of all this.

② Tool Developers:

Design and implement software tools facilitating database system design, performance monitoring, creation of GUI, prototyping etc.

③ Operators and maintenance personnel:

- Responsible for day-to-day operation of the system.

Brief Introduction of Data Models

(4) i) Explain data model & list the data models used. (3M)
ii) What is a data model? Discuss various data models. (8M)
iii) What are different datamodels present? Explain briefly. (8M)

(A) **Data Models:-**

A data model is a collection of concepts that can be used to describe the structure of a database and provides the necessary means to achieve this abstraction. whereas structure of the database means data types, relationships and constraints that should hold on the data.

Types of DataModels:-

Data Models are categorized into three types.

1. Object-based Logical Models
2. Record-based Logical Models
3. Physical Models.

1. Object-based Logical Models:-

They are used in describing data at the logical and view levels. They are ~~cht~~ characterized by the fact that they provide fairly flexible structuring capabilities and allow data constraints to be specified explicitly. The data is stored in object form. There are many different models and more are likely to come

1. The ER Model
2. The Object oriented Model
3. The Semantic data Model
4. The Functional data Model.

tutorialsDuniya.com @codes.learning

1. ER Model :-

- Entity Relationship Model is based on perception of a real worker that consists of a collection of basic objects called Entities and Relationships among these objects.
- The structure of database can be expressed by an ER digram.

→ Entity is an object / Group of objects is an Entity

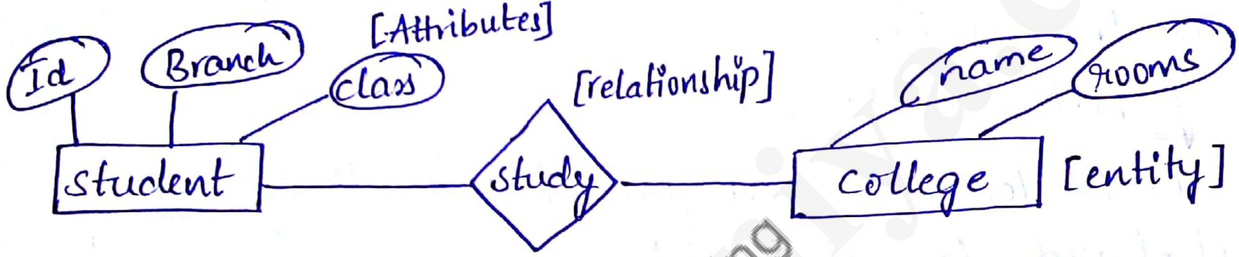
→ 'Rectangles' represent entity set.

→ Characteristics of Entities is an Attribute

→ 'Ellipses' represent Attributes.

→ 'Diamond' represent relationship among entities.

Ex:-



2. Object-Oriented Model :-

- Like ER Model, the object oriented model is based on a collection of objects. Object contains values stored in instance variables within the object.
- Classes :- collection of objects which consist of same type of values and methods.

3. Semantic Model :-

- These include the extended relational, semantic network and functional models.
- These are characterized by richer facilities for capturing the meaning of databbjects and maintaing database integrity

4. Functional Model :-

- In this Model, the operations are performed based on the predefined functions.

2. Record-based Logical Models:-

- Record based Logical Models are also used in describing data at the logical and view levels.
- Record-based models are so named because the database is structured in fixed-format records of several types. Each record type defines a fixed number of fields, or attributes, and each field is usually of a fixed length.

1. Relational Model
2. Network Model
3. Hierarchical Model.

1. Relational Model:-

- Relational Model uses a collection of tables to represent both data and relationships among those data.

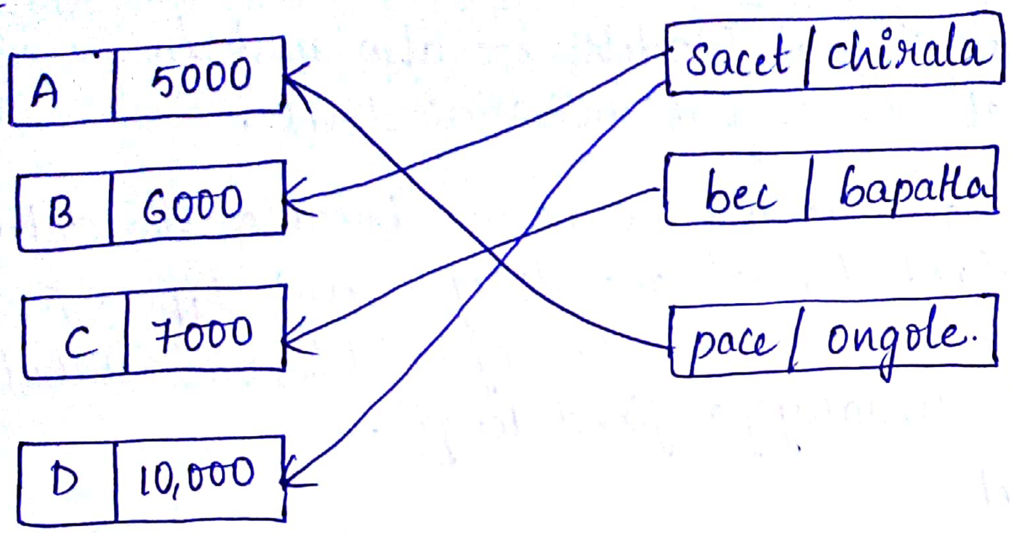
Ex:- Student Table.

St_Id	St_Name	St_Branch	St_section	st_year
5C1	P. Swathi	CSE	C	3
5C2	P. Ajay	CSE	C	3
5C3	P. Deepika	CSE	C	3
5C4	P. Rajeswari	CSE	C	3
5C5	P. Pavankalyan	CSE	C	3.

2. Network Model:-

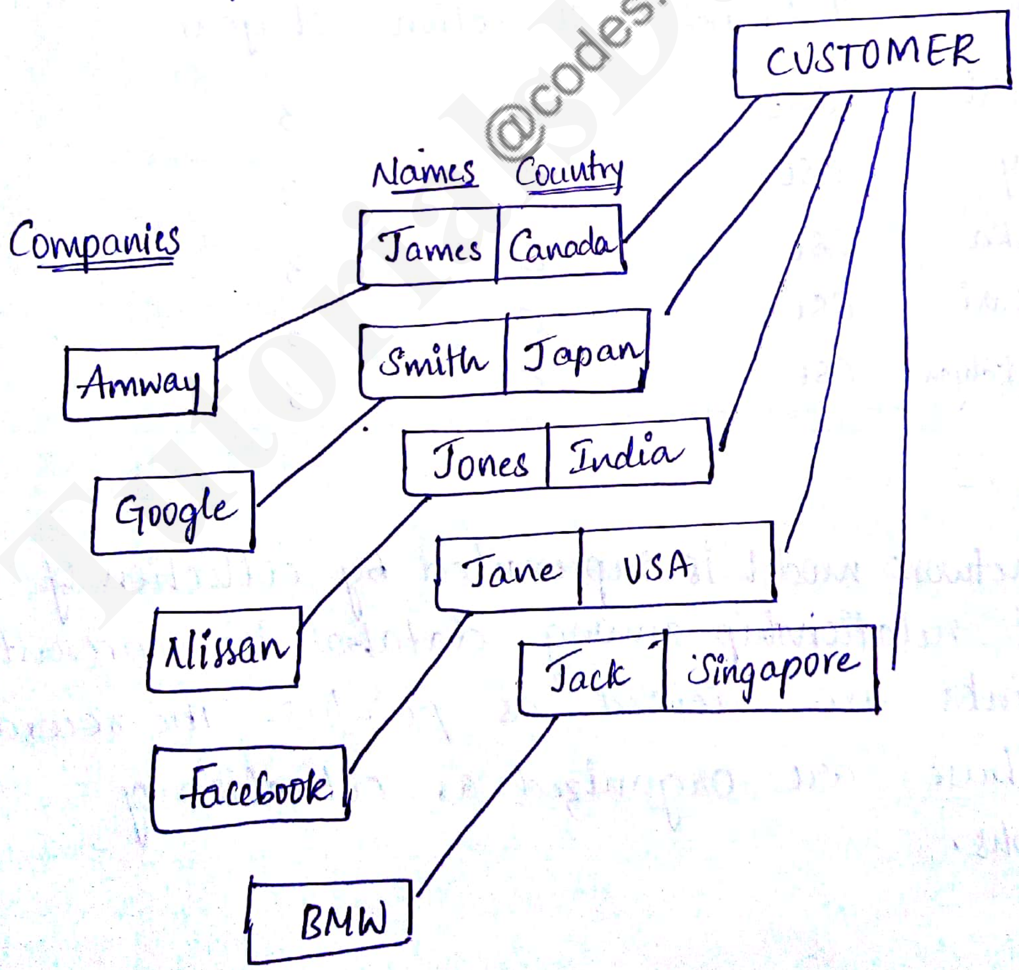
- Data in the network model is represented by collection of records, and relationship among databas is represented by links. Links are viewed as pointers. The records in the database are organized as collection of arbitary graphs.

Ex:-


A
5000
B
6000
C
7000
D
10,000
Sacet
chirala
bec
bapatla
pace
ongole.

3. Hierarchical Model:-

The Hierarchical Model is similar to the network model in the sense that data and relationships are represented by records and links. It differs from network model, in that records are organized as collection of trees rather than arbitary graphs.


CUSTOMER
Names
Country
Companies
James
Canada
Smith
Japan
Jones
India
Jane
USA
Jack
Singapore
Amway
Google
Nissan
Facebook
BMW

Advantages: ~~of~~ Hierarchical Model :-

- Simplicity
- Data Security
- Data Integrity
- Efficiency.

Disadvantages:

- Implementation complexity
- Database Management problems
- Lack of structural independence
- Programming complexity
- Implementation Limitation

4. Physical Data Models :-

Physical data Models are used to describe data at the lowest level. In contrast to logical data models, there are few physical data models in use. Two of the widely known ones are the unifying model and the frame-memory model. Here, the data is stored in Transperancy.

Q. NO – ⑦

Concept:- Three tier schema Architecture

FAQ'S:-

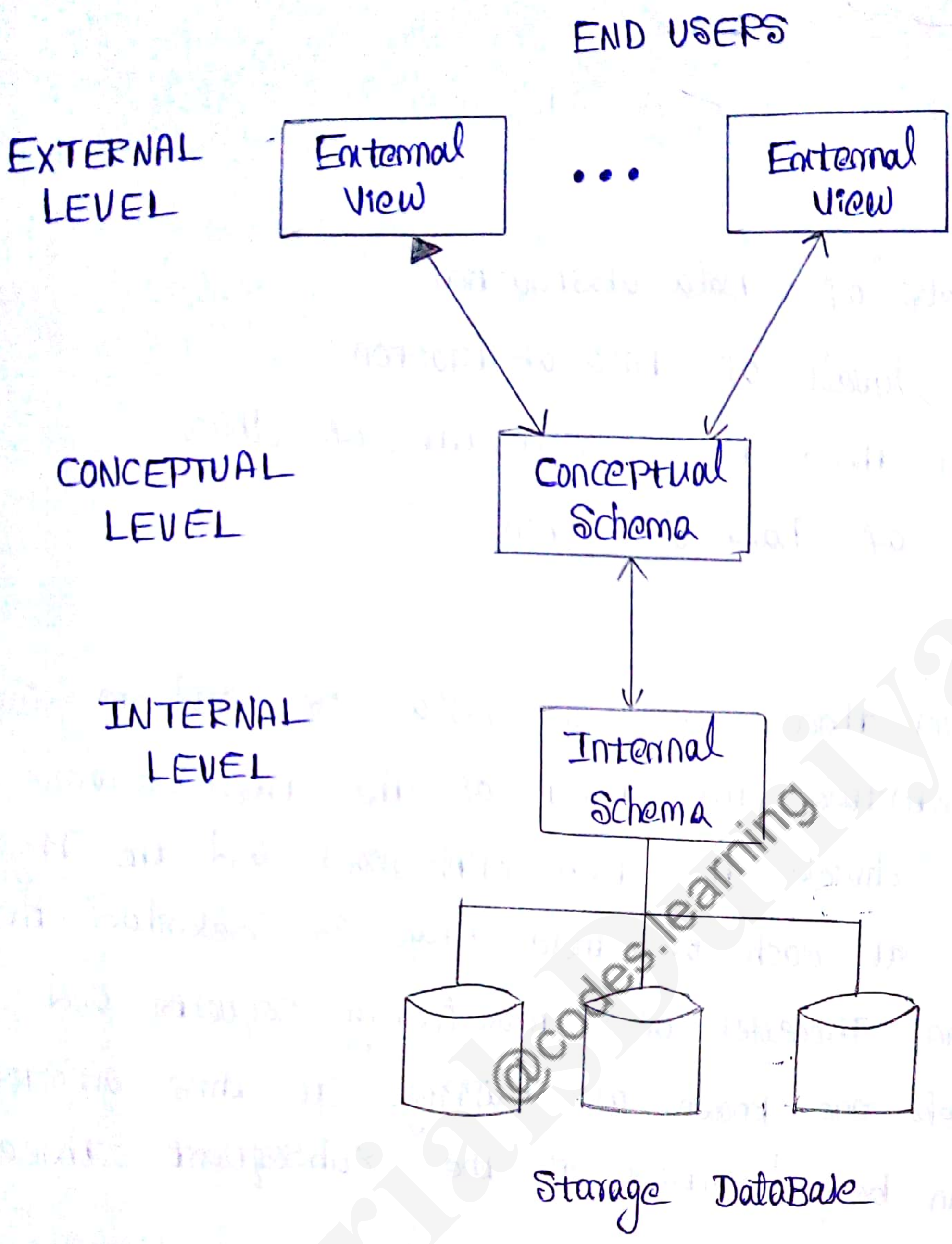
1. Give the levels of Data abstraction.
2. What are the levels of Data abstraction
3. Draw & explain three tier Architecture of dbms.
4. Different ways of Data abstraction.

Theory:-

A DBMS gives three levels of data is said to follow three-level architecture. The target of the three-schema architecture is to divide the user applications and the Physical database. The view at each of these stages is described through a schema. The Processes of transforming requests and results among levels are known as mappings. In this architecture, schemas can be described at the subsequent three stages.

External level or Subschema:-

It is the highest level of database abstraction whereas only those Portions of the database of concern to a user or application Program are involved. Any number of user views (Some of which may be identical) might exist for a given global or conceptual view Each external view is described through means of a Schema known as External or Sub Schema.


END USERS
EXTERNAL LEVEL
External View
• • •
External View
CONCEPTUAL LEVEL
Conceptual Schema
INTERNAL LEVEL
Internal Schema
Storage DataBase

Conceptual Level or conceptual Schema :-

At these stages of database abstraction all the database entities and the relationships between them are included. one conceptual view represents the whole database. This conceptual view is described through the conceptual schema. There is only one conceptual schema per database. The description of data at that levels in a format independent of its Physical representation. It is also involves features which specify the checks to retain data consistency and Integrity.

Internal level or Physical Schema:-

It is closest to the Physical Storage technique used. It denotes how the data will be stored and elaborates the data structures and access techniques to be used through the database. The internal view is expressed through the internal schema.

9. CONCEPT:- DATA BASE ENVIRONMENT.

FAQ's:- 1. what do you mean by environment in database systems? Explain with data base system architecture (8M)

Data Base Environment:-

One of the main aims of database is to supply users with an abstract view of data, hiding certain element of how data is stored and manipulated. so, the starting point for the design of a database must be an abstract and general description of the information requirements of the organisation that is to be represented in database. And hence you will require an environment to store data and make it work like a database. Now we are going to know about database environment and architecture.

What is a database Environment?:-

- A data base environment is a collective system of components that comprises and regulates the group of data, management and use of data which consists of

(i) software
(ii) hardware
(iii) people
(iv) Techniques of handling database (procedures)
(v) data.

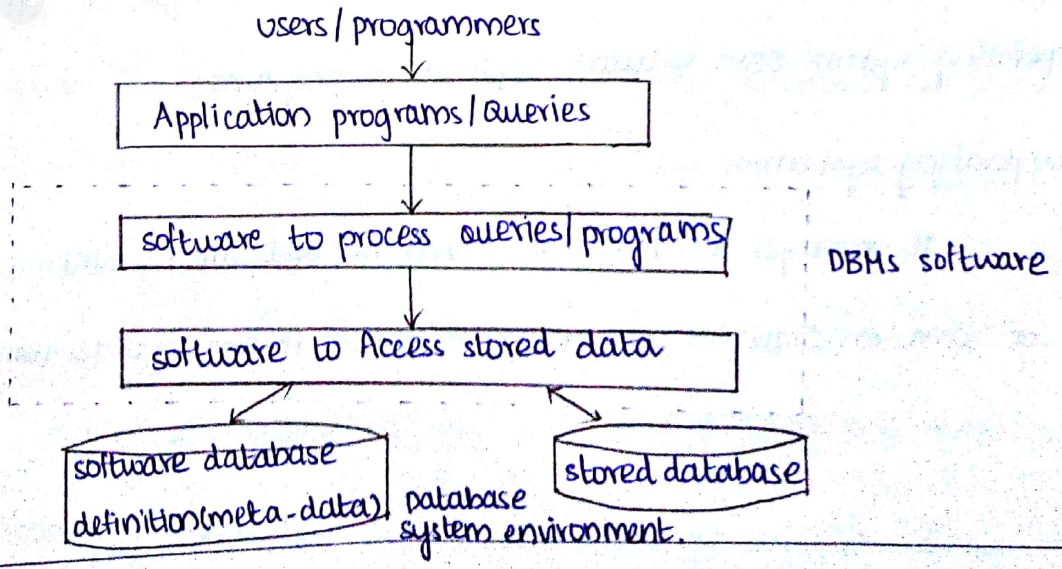

users / programmers
Application programs / Queries
software to process queries / programs
DBMS software
software to Access stored data
software database definition (meta-data)
stored database
Database system environment.

Fig:-1.

Fig1. shows outline of database system environment.

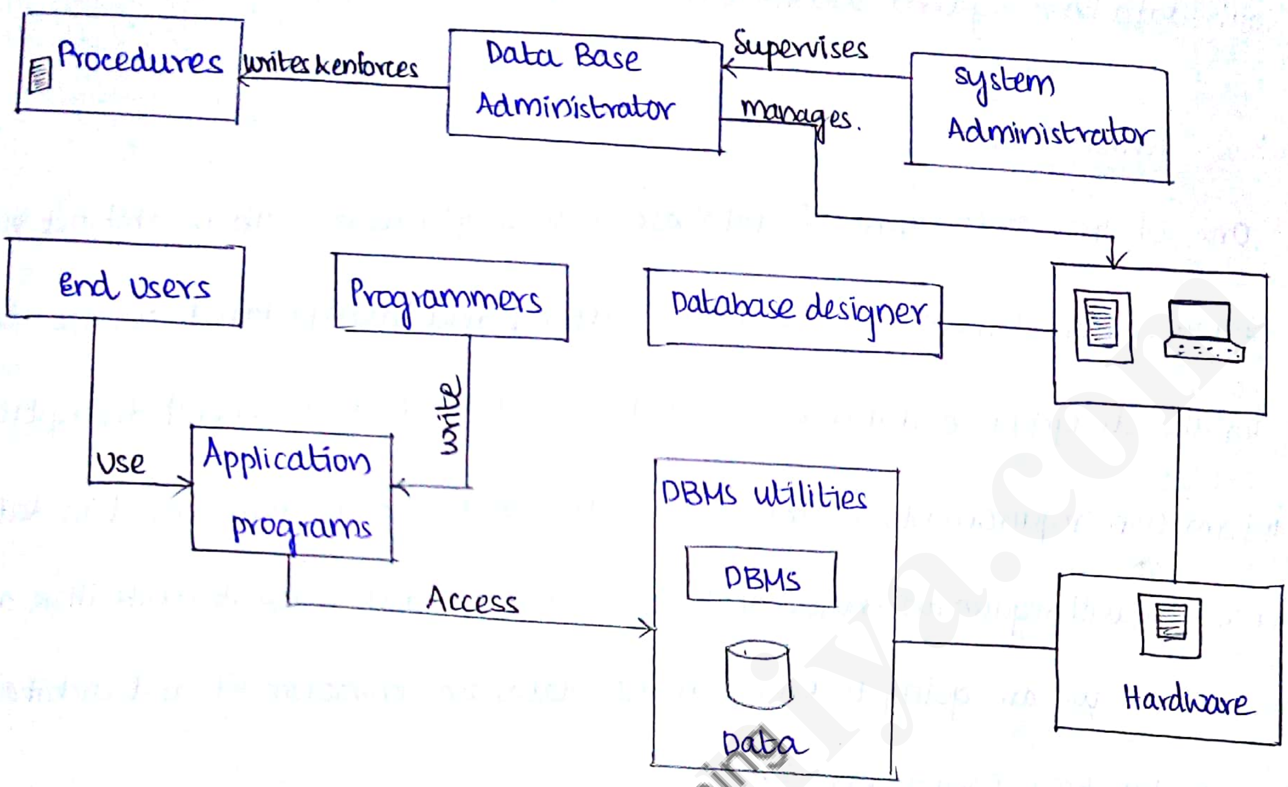

Procedures
writes & enforces
Data Base Administrator
Supervises
manages.
System Administrator
End users
Programmers
Database designer
Use
Application programs
write
DBMS utilities
DBMS
Access
Data
Hardware

Fig2: Detailed View of Database Environment.

1. Hardware:-

Hardware refers to all system's peripheral devices; for example, computers storage devices, printers, network devices and etc.

2. software:-

To make database system work properly, three types of software are needed: operating systems, DBMS software, application programs.

a) operating systems:-

It manages all hardware components and allows other software to run on computers. Examples of operating systems software include windows, Linux etc.

b) DBMS software :-

It manages database within the database system. some examples of DBMS software include oracle, Access, Mysql and etc.

c) Application programs:-

These are used to access and manipulate data in DBMS and to manage the computer environment in which data acess and manipulation takes place. Application programs are most commonly used to access data to generate reports. Most of the application programs provide GUI.

3. People:-

This component includes all users of database system. According to job nature, five types of users can be identified : system administrators, database administrators, database designers, system analysts and programmers, and end users.

a) system Administrators:-

They supervise the database system's general operations.

b) Database Administrators:-

They are also known as DBAs. They manage the DBMS and ensure that database is functioning properly.

c) DataBase Designers:-

They design the database structure. They are the database architects. As this is very critical, the designer's job responsibilities are increased.

d) system Analysts and programmers:-

They design and implement the application programs. They design & create data entry screens, reports and procedures through which end users can access & manipulate data.

TutorialsDuniya.com @codes.learning

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

@codes.learning

e) End Users:-

They are the people who use the application programs to run the organisation's daily operations. For example, sales-clerks, supervisors, managers are classified as end users.

4. Procedures:-

Procedures are the instructions and rules that supervise the design and use of database system. procedures are a critical component of the system. procedures play an important role in a company because they enfore the standards by which business is conducted in an organisation.

5. Data:-

Data refers to collection of facts stored in database. Because data are the raw material from which information is generated, no database can exist without data.

Additional Questions:-

1. Explain history of database systems. (4M)

A. Ancient Times:-

RAM was expensive and limited. programmer productivity Low. Programmer defined both logical and physical structure, such as storage structure, access methods, I/O modes etc.

1950-1960's:-

Data maintained in flat file. processing characteristics determined by common use of magnetic tape medium.

1960 – 1970's:-

First DBMS, hierarchical DBMS introduced by IBM. This type of DBMs based on binary trees, where the shape was like a tree and relations were only limited between parent and child records.

Benefits:- less redundant data, data independence, security and integrity.

Drawback:- complex implementation

Next, network DBMS introduced by charles Bachmann at Honeywell. In this model, each record can have multiple parents in comparision with one in hierarchical DBMS.

Drawback:- Difficulty in design and maintenance.

1970 – 1990's:-

Relational DBMs, introduced by edger codd. This was a new system for entering data and working with big databases, where the idea was to use a table of records. All the tables will be linked by one to one, one to many, many to many relationships.

Benifit:- High performance in transaction processing.

2. Define different characteristics of database systems. (3M (or) 8M)

A. **characteristics:-**

1. **Real-world Entity**

DBMS uses real world entities to design its architecture. It uses behaviour & attributes too.

Eg:- school database use students as entity & age as an attribute.

2. **Relation-based Tables:-**

DBMs allows entities and relations among them to form tables. User can understand architecture just by looking at table names.

3. Isolation of data and Application

Database is different from data. Database is an active activity whereas data is said to be passive. DBMS uses metadata, to ease its own process.

4. Less Redundancy

By following the rules of normalization, redundancy is reduced.

5. consistency

6. Query Language.

7. ACID properties.

8. Multiuser and concurrent Access.

9. Multiple views.

10. security.

3. List different types of database users and activities of all different database users. (4M)

A. There are four different types of database users based on how they interact with the system.

(i) Application programmers.

(ii) sophisticated Users. }
(iii) specialized users. } End Users.
(iv) Native Users. }

(i) Application Programmers:-

They are computer professionals who interact with the system through DML calls which are embedded in a program written in a host language (for eg:- COBOL, C). Since DML syntax is different from host language syntax, DML calls are usually prefaced by a special character so that appropriate code can be generated.

A special processor called DML precompiler, converts the DML statements to normal procedure calls in host language.

2. Sophisticated Users:-

They interact with systems without writing code. Instedd, they form their request in a database query language. Each subquery is submitted to a query processor whose function is to break down DML statement into instructions that the storage manager understands.

3. Specialized Users:-

specialized users are sophisticated users who write specialized database applications that do not fit into the traditional data processing framework. Among these applications are computer aided design systems, knowledge-base & expert systems.

4. Naive Users:-

They are unsophisticated Users who interact with the system by invoking one of the permanent application programs that have been written previously.

Centralized and Client-Server DBMS Architectures:

Centralized DBMS:

- It combines everything into single system including DBMS software, hardware, application programs, and user interface processing software.
- User can still connect through a remote terminal however, all processing is done at centralized site.

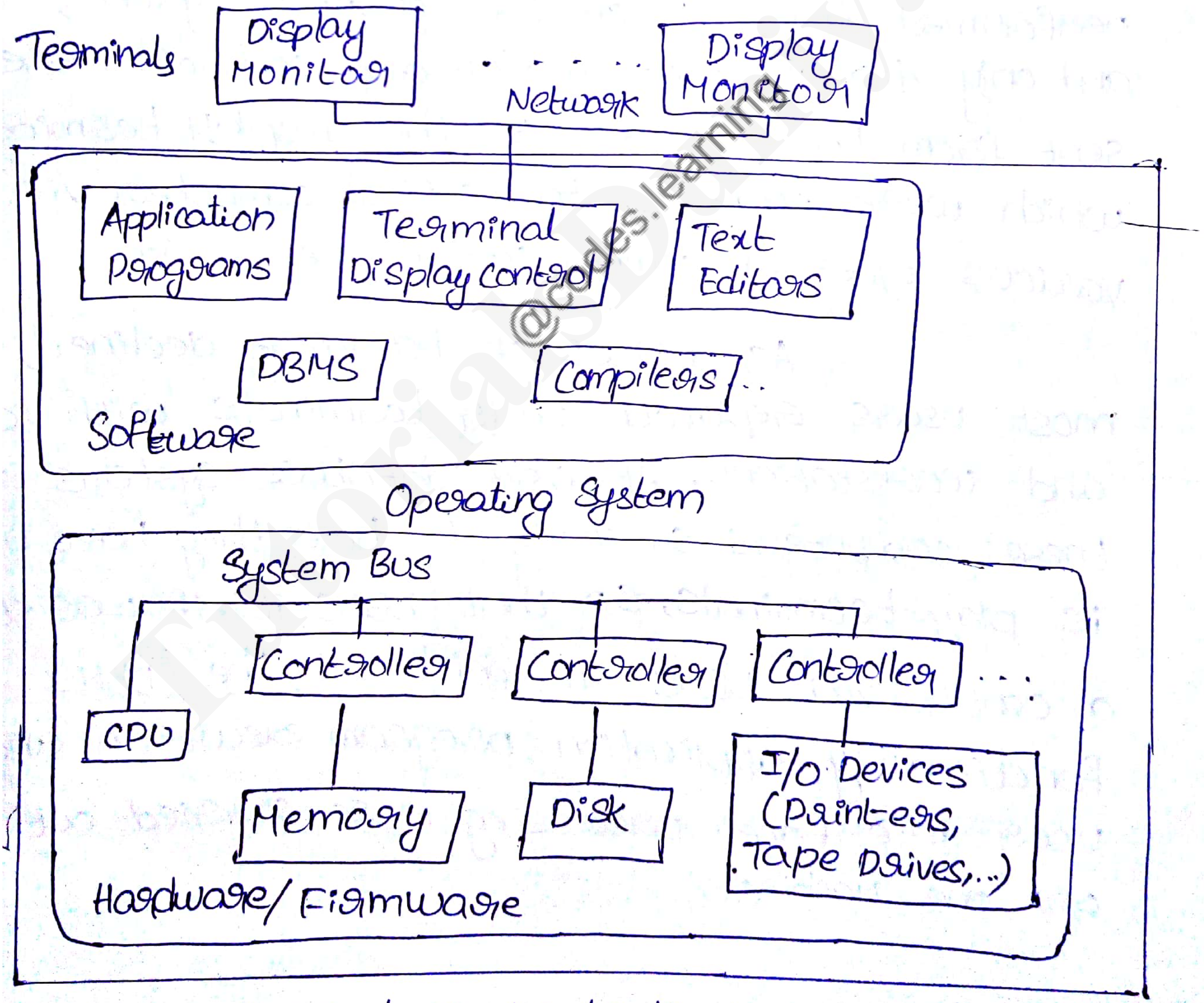

Terminals
Display Monitor
Display Monitor
Network
Application Programs
Terminal Display control
Text Editors
DBMS
Compilers
Software
Operating System
System Bus
Controller
Controller
Controller
CPU
Memory
Disk
I/O Devices (Printers, Tape Drives,..)
Hardware/Firmware

Fig: A physical centralized architecture

Architectures for DBMS have followed trends similar to those generating computer system architectures. Earlier architectures used mainframes computers to provide the main processing for all system functions, including user application programs and user interface programs as well as DBMS functionality. The reason was that most users accessed such systems via computer terminals that did not have processing power and only provided display capabilities. Therefore all processing was performed remotely on the computer system, and only display information and controls were sent from the computer to the display terminals, which were connected to central computer via various types of communication networks.

As prices of hardware declined, most users replaced their terminals with PCs and workstations. At first database systems used these computers similarly to how they have used is play terminals, so that DBMS itself was still a centralized DBMS in which all the DBMS functionality, application program execution and user interface processing were carried out on one Machine.

Basic 2-tier Client-Server Architecture:

- Specialized servers with specialized functions
- Print server
- File server
- DBMS server
- Web server
- Email server
- clients can access the specialized servers as needed.

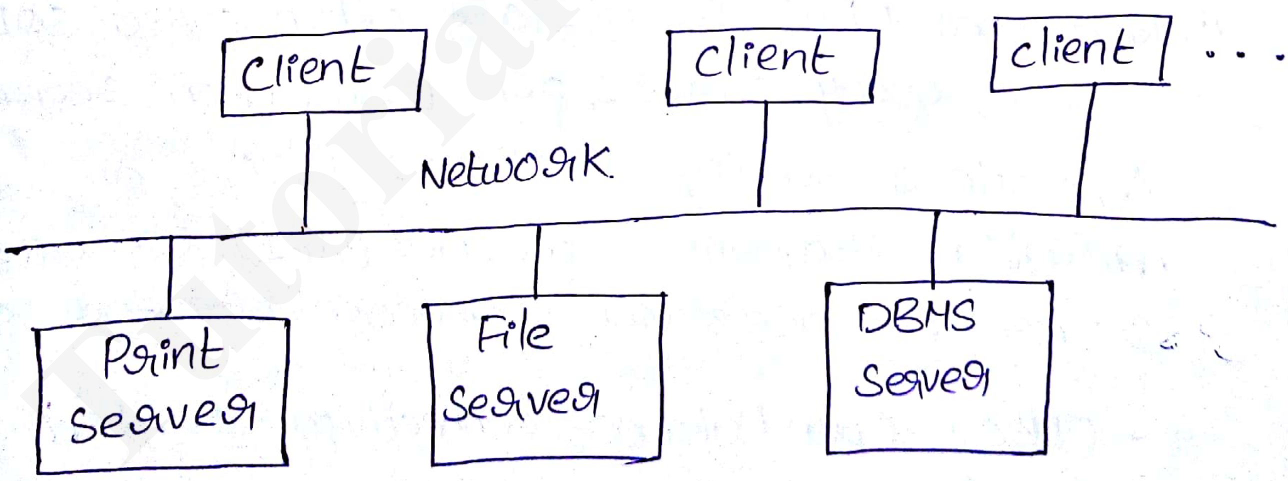

Client
Client
client
. . .
Network
Print server
File server
DBMS server

Fig: Logical two-tier client/server architecture

Clients:

i, Clients provide appropriate interfaces through a client software module to access and utilize the various server resources.

ii, Clients may be diskless machines or PCs or workstations with disks with only the client softwared installed.

iii, Connected to the servers via some form of a network like local area network (LAN), Wireless network etc..

DBMS server:

i, It provides database query and transaction services to the clients.

ii, Relational DBMS servers are often called SQL servers, query servers, or transaction servers. Applications running on clients utilize an Application Program Interface (API) to access server databases via standard interface such

- ODBC : Open Database connectivity standard.
- JDBC: for Java programming access

iii, client and server must install appropriate client module and server module software for ODBC or JDBC

Two Tier Client-Server architecture:

- A client program may connect to several DBMSs, sometimes called the data sources.
- In general, data sources can be files or other non-DBMS software that manages data. Other variations of clients are possible.

  Ex: In some object DBMSs, more functionality is transferred to clients including data dictionary functions, optimization and recovery across multiple servers etc.

Three Tier client-server architecture:

- Common for web applications.
- Intermediate layer called Application Server or Web Server.
- It stores the web connectivity software and the business logic part of the application used to access the corresponding data from the database server.
- It acts like a conduit for sending partially processed data between the database server and the client.

- Three-tier Architecture can enhance security.
- Database server only accessible via middle tier
- clients cannot directly access database server.

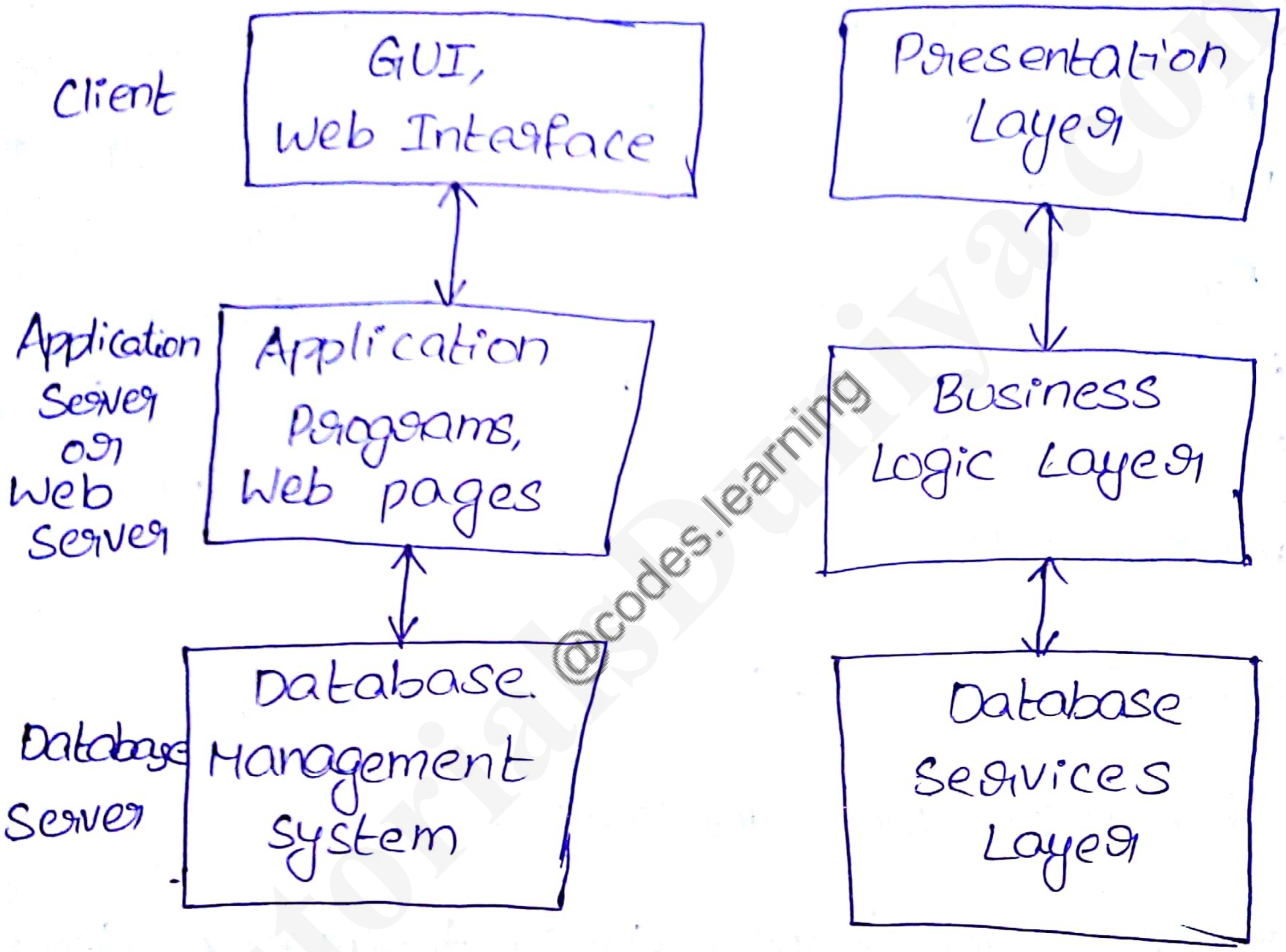
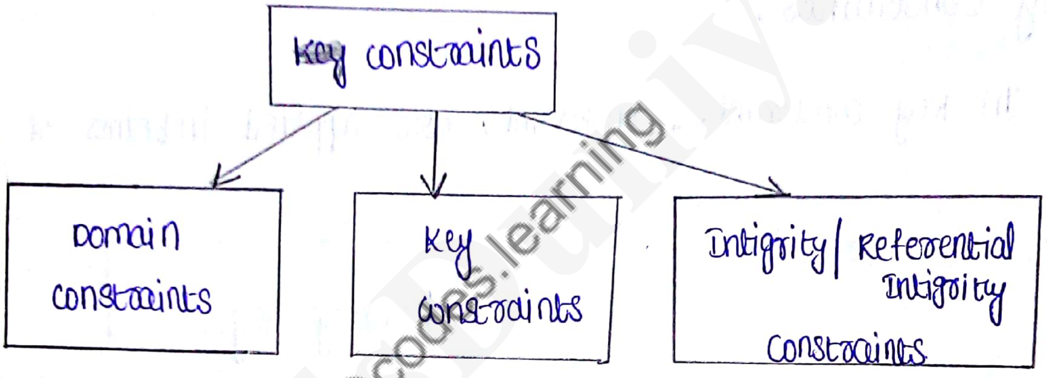

Client
GUI, Web Interface
Application Server or Web Server
Application Programs, Web pages
Database Server
Database Management System
Presentation Layer
Business Logic Layer
Database Services Layer

Fig: Logical three-tier client/server architecture.

Key Constraints:

A key constraint is a statement that a certain minimal subset of the fields of a relation is a unique identifier for a tuple. A set of fields that uniquely identifies a tuple according to a key constraint.


Key constraints
Domain constraints
Key constraints
Intigrity / Referential Intigrity constraints

(1) **Domain Constraints**: This domain constraints are applied a particular field.

* CHECK key is one of the Domain key.

CHECK constraint:

Its a check constraint is a type of Domain constraint in SQL which specifies a requirement that must be met by each row in a database table.

- It is used to check the condition.

Example:

```
create domaine D1
 sal float
  check sal_value >= 20,000;
```

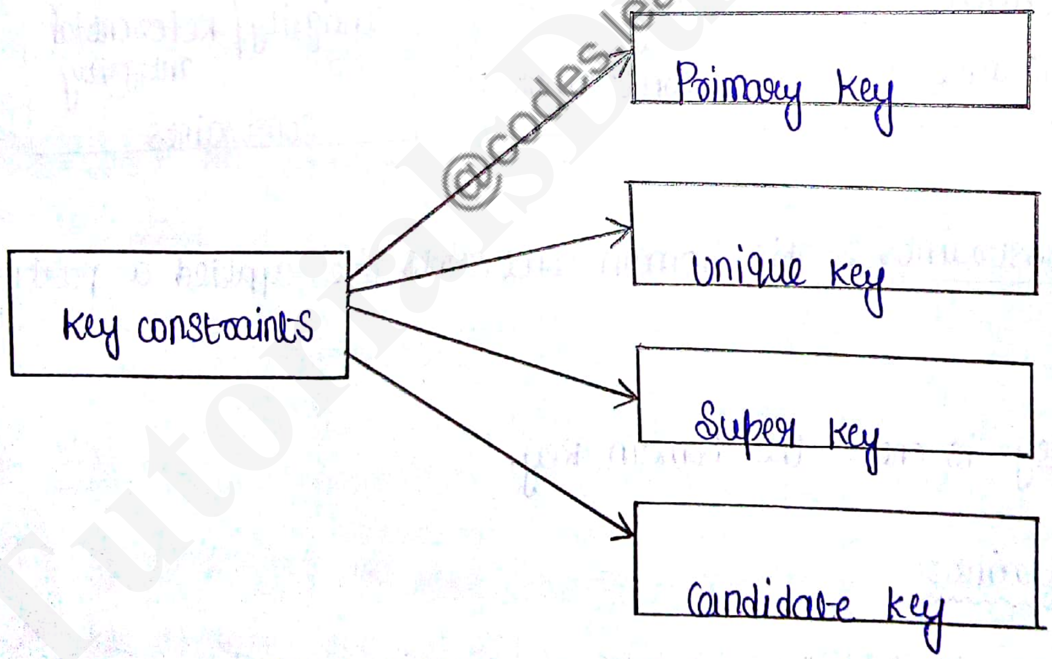
Checks conditions as follows by using CHCK:

1. First check salary in float (or) not
2. second check enter sal_value >20,000 (or) not.

If you enter sal_value < 20,000 it generate's an error.

(2) Key Constraints:

In key constraints, constraints are applied in terms of keys.



(a) Primary key: A primary key is a field in a table which uniquely identifies each row/record in a database table. primary keys must contain unique values. A primary key column cannot have duplicate values and Null values.

@codeslearning TutorialsDuniya.com

https://www.tutorialsduniya.com Scanned by CamScanner

characteristics of a primary key:

- * one per base table
- * cannot be null
- * Is defined as a table constraint
- * Best practice is a single column

usage: create table student(sid integer PRIMARY KEY, Name varchar2(20), class integer, Age integer);

SID	NAME	class	AGE

Not allow duplicate and NULL values because of SID containe applied by primary key.

prime attribute: An attribute containe primary key constraint that attribute is called as prime attribute.

Example: SID (in above table)

(b) Unique Key:

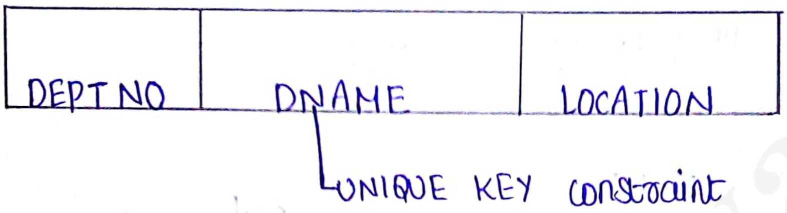
unique key constraints are used to ensure that data is not duplicate in two rows in the database. one row in the database is allowed to have null for the value of unique key constraint.

characteristics of a unique key:

- * one or more per base table
- * can be null
- * one or more columns
- * Any data type

- It is typically a known value (i.e. it is not hidden)
- May be used to establish referential constraints.

Example : create table dept(dept_no integer, dname varchar2(20) UNIQUE,
location varchar2(20));

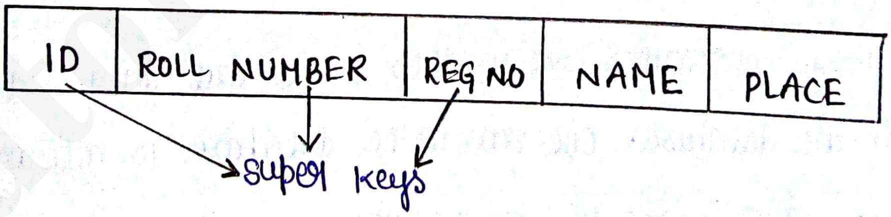

DEPT NO
DNAME
LOCATION
UNIQUE KEY constraint

(no row may duplicate a value in the constraint's column)

(c) Super key : An attribute or a combination of attribute that is used to identify the records uniquely is known as super key.

- A table have many super keys.
- It is also allow duplicate values also.

Examples :


ID
ROLL NUMBER
REG NO
NAME
PLACE
super keys

(1) {ID, ROLL NUMBER, REG NO}
(2) {ID, PLACE, REG NO}
(3) {ID, NAME, REG NO}
.....

} super keys Examples.

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

@codes.learning

(d) Candidate Key :

candidate key can be defined as minimal super key or irreducible super key. In other words an attribute or combination of attributes that identifies the record uniquely but none of its proper subsets can identify the records uniquely.

Example :

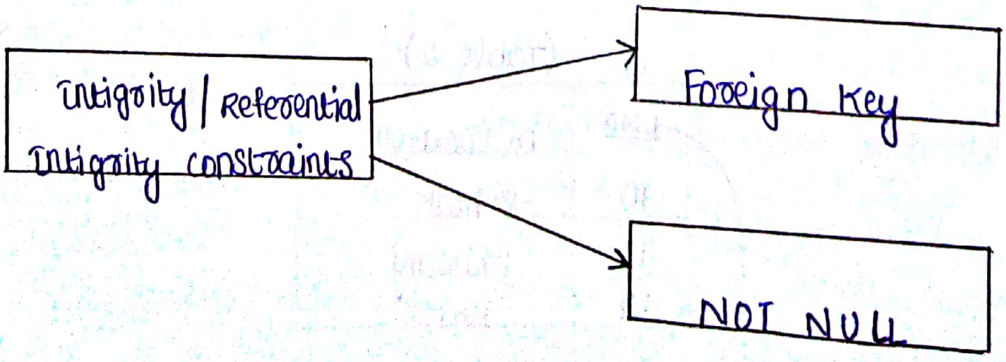
STUID	FIRST NAME	LAST NAME	COURSE

{STUID}
{FIRSTNAME, LAST NAME}
{ STUID, COURSE}
} examples of candidate key.

→ primary key , UNIQUE keys are used practically but not candidate key and super key.

3. Intigrity constraints :

Intigrity constraints means depending upon constraints enter correct data or not. That's means entered data as per the constraints (or) not.


Intigrity / Referential Intigrity constraints
Foreign key
NOT NULL

(a) Foreign key:

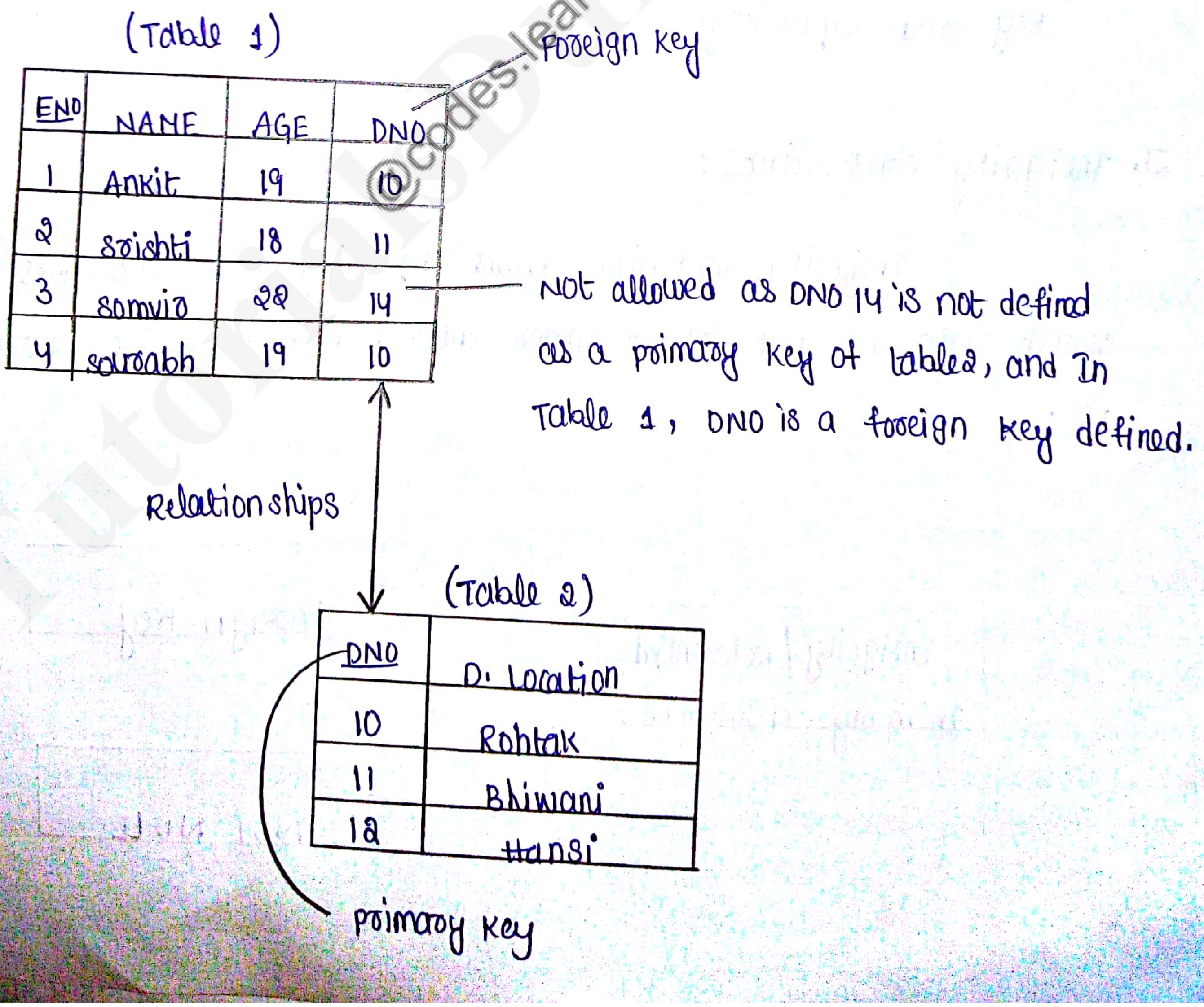
It is used to enforce referential integrity between tables in a relational database. primary key field cann't contain null values but foreign key field can contain null values.

Foreign key constraint can be defined by a CREATE TABLE statement or an ALTER TABLE statement.

characteristics of a foreign key:

- * one or more per base table
- * can be null
- * one or more columns corresponding to like columns in the parent table.

Example:



(Table 1)

ENO	NAME	AGE	DNO
1	Ankit	19	10
2	Srishti	18	11
3	Somvir	22	14
4	Saurabh	19	10

Not allowed as DNO 14 is not defined as a primary key of table2, and In Table 1, DNO is a foreign key defined.

(Table 2)

DNO	D. Location
10	Rohtak
11	Bhiwani
12	Hansi

(b) **NOT NULL:**

The NOT NULL constraint enforces a column to NOT accept NULL values. This enforces a field to always contain a value, which means that you can't insert a new record, or update a record without adding a value to this field.

Example: create table student(sid integer NOT NULL, Name varchar2(20), Age integer);

The above query will declare the sid field of student table will not take NULL value.

Another key constraints:

1. Composite key:

If use multiple attributes to create a primary key then that primary key is called composite key. That means set of superkeys and candidate keys.

2. Alternate key:

Alternate key can be any of the candidate keys except for the primary key.

Example: "Name, Address" as it is only other candidate key which is not a primary key.

Basic SQL querying (select & project) using where clause.

SQL allows a table (relation) to have two or more tuples that are identical in all their attributes values. Hence, an SQL table is not a set of tuple, because a set does not allow two identical numbers; rather ~~it~~ it multiset of tuples.

A basic query statement in SQL is the SELECT statement.

The SELECT statement used in SQL has no relationship to the SELECT operation of relational algebra.

The SELECT statement:-

The syntax of this command is:

```
SELECT <attribute list>
FROM   <table list>
WHERE  <condition>;
```

Query 0: Retrieve the birthday and address of employee(s) whose name is 'John B. Smith'

```
Q0 : SELECT BDATE, ADDRESS
     FROM EMPLOYEE
     WHERE FNAME = 'John' AND MINIT='B' AND LNAME='SMITH'
```

Query : Retrieve the name and address of all employee who work for the Research dept.

```
Q: SELET FNAME, LNAME, ADDRESS
   FROM EMPLOYEE, DEPARTMENT
   WHERE DNAME = 'Research' AND DNUMBER=DNO
```

1. Relational model, concepts of domain, attributes, ~~typle~~ tuple, relation, importance of Null values.

A) Relational Model :

R(A1, A2,----, An) is a relational schema of degree n denoting that there is a relation R having as its attributes A1, A2, ---- An.

By convention, Q, R and S denote relation names.

By convention, q, r and s denoto relation states. For example, r(R) denotes one possible state of relation R. If R is understood from context, this could be written, more simply, as r.

By convention, t, u, and v denote tuples.

the "dot notation" R.A (Eg, STUDENT.Name) is used to qualify an attribute name, ~~used~~ usually For the purpose of distinguishing it from a same-named attribute in a different relation (Eg. DEPARTMENT.Name).

Domain concepts :- .

A (usually named) set/universe of atomic values, where by "atomic" we mean simply that, from the point of view of the database, each value in the domain is indivisible (i-e cannot be broken down into component parts).

Examples of domains:

USA_phone_number: string of digits of length ten

SSN: string of digits of length nine

Name: String of characters beginning with an upper case letter

GPA: a real number between 0.0 and 4.0.

Sex: a member of the set {female, male}

Dept-code: a member of the set {CMPS, MATH, ENGL PHYS, PSYC, ...}

These are all logical descriptions of domains. For implementation purposes, it is necessary to provide descriptions of domains in terms of concrete data types (or formats) that are provided by the DBMS (such as string, int, boolean), in a manner analogous to how programming languages have intrinsic data types.

Attributes: The name of the role played by some value (coming from some domain) in the context of a relational schema. The domain of attribute A is denoted dom(A).

Tuple: A tuple is a mapping from attributes to values drawn from the respective domains of those attributes. A tuple is intended to describe some entity (or relationship between entities) in the miniworld.

As an example, a tuple for a PERSON entity might be

{ Name ---> "Rumpelstiltskin", Sex ---> Male,
IQ → 143 }.

Relation: A (named) set of tuples all of the same form (i.e. having the same set of attributes). The term table is a loose synonym. (some database purists would ~~agree~~ argue that a table is "only" a physical manifes-tation of a relation).

Importance of NULL values:

Every column in a table should contain a value, through there may be times when the value is unknown

For example, consider the following table which stores data relating to stories on CD supplies.

SUPPLIER

supplierid	Name	Address	phone	Fax

columns of the SUPPLIER table.

1) supplier id
2) supplier name
3) Address
4) supplier phone
5) supplier Fax.

To communicate with suppliers you will need thei name, Address, phone number, and fax. If you do not know one or more of those pieces of data, you will not know that to enter into its corresponding column.

A zero, by constrast, is an INT or DECIMAL value. If a stories on CD supplier gave the company a thousand customers who placed an order, the Retail price column for that CD would contain a zero.

List SQL Grouping functions with examples & explain each function with example?

What is SQL single row function? By means of suitable examples illustrate the usage of SQL date, character and number functions.

Arithmetic & Logical Operations in Basic SQL:-

Arithmetic operators can perform arithmetic operations on numeric operands involved. Arithmetic operators are -

1. Addition (+)
2. subtraction (-)
3. Multiplication (*)
4. division (/).

The addition (+) and subtraction (-) operators can also be used in date arthimetic functions.

Operator	Meaning
+ (add)	Addition
- (subtract)	Subtraction
* (multiply)	Multiplication
/ (divide)	division
% (Modulo)	Modulo division.

Syntax:-

```
select   <expression1> [arithmetic operator]<expression2> ...
from     table_name
where    [expression/condition];
```

SQL plus(+) operator:-

The SQL plus (+) operator is used to add two or more expressions or numbers. SQL plus(+) operator is nothing but add(+) operator.

Example:①

```
Select 15+10 from dual;
```

O/P:- 25

Example:②

Sample table: Student

Student_name	marks1	marks2	marks3	marks4
Ram	90	91	92	93
Ravi	80	81	82	83
Raju	70	71	72	73
Revanth	91	92	93	94

The following can be used to get the total marks of every student.

```
select student_name, marks1, marks2, marks3, marks4, (marks1 + marks2+marks3
+marks4) as total
from student;
```

Output:-

```
Student_name   marks1   marks2   marks3   marks4   total
------------   ------   ------   ------   ------   -----
Ram            90       91       92       93       366
Ravi           80       81       82       83       326
Raju           70       71       72       73       286
Revant         91       92       93       94       370

4 rows selected.
```

2. SQL Minus (-) operator :-

The SQL Minus (-) operator is used to subtract one expression or number from another expression or number. SQL Minus (-) operator is nothing but subtraction (-) operator.

Example ①

```
select 15-10 from dual;
```

O/P :- 5

Example②

Sample table: Bank_account.

Account_holder	Total_Amount	withdraw_amount
Ram	10,000	5,000
Ravi	15,000	10,000
Raju	20,000	15,000
Rajesh	25,000	20,000

The following command can be used to get the remaining amount of each account holder.

```
select Account_holder, Total_Amount, withdraw_Amount, (TotalAmount - withdraw_amo
unt) as remaining_amount
from Bank_account;
```

output:-

Account_holder	Total_Amount	withdraw_Amount	remaining_amount
Ram	10,000	5,000	5,000
Ravi	15,000	10,000	5,000
Raju	20,000	15,000	5,000
Rajesh	25,000	20,000	5,000

3SQL multiply(*) operator:-

The SQL multiply (*) operator is used to multiply two or more expressions or numbers.

Example①

```
select 15*10 from dual;
```

O/P:- 150

Example②

Sample table : ~~Business-profit~~ student_percentage.

student_name	percentage	Totalmarks
Ravi	91.6	600
Ram	92.6	600
Raju	93.6	600
Rajesh	94.6	600

To get the total marks attained by every student, we are using the following command.

```
select student-name, percentage, totalmarks, (percentage x totalmarks) as attaine
marks
from student-percentage;
```

output:-

student_name	percentage	Total marks	attained_marks = $\frac{\text{Percent} \times \text{Total}}{100}$
Ravi	91.6	600	550
Ram	92.6	600	~~490~~ 556
Raju	93.6	600	562
Rajesh	94.6.	600	568.

4 rows selected.

4. SQL divide (/) operator:-

The SQL divide (/) operator is used to divide one expression or number by another expression or number.

Example①

```
select 15/10 from dual;
```

O/P :- 1.5

Example②

Sample table: student

student_name	marks1	marks2	marks3	total
Ram	90	91	92	273
Ravi	80	81	82	243
Raju	70	71	72	213
Rajesh	90	92	93	275

To get the average marks we are using the following command.

```
select student_name, marks1, marks2, marks3, total, (total/3) as average
from ~~dual;~~ student;
```

Output:-

```
student_name   marks1   marks2   marks3   total   average
------------   ------   ------   ------   -----   -------
Ram              90       91       92      273      91
Ravi             80       81       82      243      81
Raju             70       71       72      213      71
Rajesh           90       92       93      275      91.666

4 rows selected.
```

SQL logical Operators :-

Logical operators compare two conditions at a time to determine whether a row can be selected for the output. when retrieving data using a select statement, you can use logical operators in the where clause, which allows you to combine more than one condition. These are three logical operators. They are-

1. OR
2. AND
3. NOT

Syntax:-
select column from table_name where <condition1> [logical operator] <condition2>;

Logical operator	Description
OR	for the row to be selected at least one of conditions must be true.
AND	For a row to be selected all the specified conditions must be true.
NOT	for a row to be selected the specified condition must be false.

1. "OR" logical Operator :-

If you want to select rows that satisfy at least one of the given conditions then you can use the logical operator OR.

Example

Sample table: student

Student_name	Percentage	Result
Ram	92.2	Pass
Ravi	82.6	pass
Raju	73.6	fail
Rajesh	66.6	Pass
Revanth	70.7	fail

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

@codes.learning

4. SQL divide (/) operator:-

The SQL divide (/) operator is used to divide one expression or number by another expression or number.

Example ①

```
select 15/10 from dual;
```

O/P :- 1.5

Example ②

Sample table: Student

Student_name	marks1	marks2	marks3	total
Ram	90	91	92	273
Ravi	80	81	82	243
Raju	70	71	72	213
Rajesh	90	92	93	275

To get the average marks we are using the following command.

```
select student_name, marks1, marks2, marks3, total, (total/3) as average
from ~~dual;~~ student;
```

Output:-

```
student_name   marks1   marks2   marks3   total   average
------------   ------   ------   ------   -----   -------
Ram              90       91       92      273      91
Ravi             80       81       82      243      81
Raju             70       71       72      213      71
Rajesh           90       92       93      275      91.666

4 rows selected.
```

To get the student_name whose percentage is greater than 73 or whose result is pass, the following command is used.

```
Select   student_name
from   student
where   percentage>73  or  Result = 'pass';
```

Output:-

```
student_name
------------
Ram
Ravi
Raju
Rajesh
4 rows selected.
```

The following table describes how logical "OR" operator selects a row.

column1 Statisfied?	column2 satisfied?	Row selected?
yes	Yes	Yes
yes	NO	Yes
NO	Yes	Yes
NO	NO	NO

2. "AND" logical Operator:-

If you want to select rows that must satisfy all the given condition, you can use the logical "AND" operator.

Example:

Sample table: Student.

Student_name	percentage	result
Ram	92.2	pass
Ravi	82.6	Pass
Raju	73.6	fail
Rajesh	66.6	pass
Revanth	70.6	fail

To get the student_name whose percentage is greater than 73 and whose result is pass, the following command is used.

```
select   student_name
from     student
where    percentage>73 and result ='pass';
```

Output:

```
Student_name
------------
  Ram
  Ravi
  2 rows selected.
```

The following table describes how logical "AND" operator selects a row.

Column1 satisfied?	column2 satisfied?	Row selected
yes	yes	yes
Yes	NO	NO
NO	Yes	NO
NO	NO	NO

3. "NOT" Logical Operator:-

If you want to find rows that do not satisfy a condition, you can use the logical operator, NOT. NOT results in the reverse of a condition. That is, if a condition is satisfied then row is not returned.

Example:

Sample table: student

Student_name	percentage	Result
Ram	92.2	pass
Ravi	82.6	pass
Raju	73.6	fail
Rajesh	66.6	pass
Revant	70.6	fail

To get the student_name who did not fail, the following command is used.

```
select   student_name
from   student
where    not   Result = 'fail';
```

Output:-

```
student_name
-------------
 Ram
 Ravi
 Rajesh.
3 rows selected.
```

The following table describes how logical "NOT" operator selects a row.

column1 satisfied?	NOT column1 satisfied?	Rowselected
yes	NO	NO
NO	yes	yes.

SQL FUNCTIONS:-

Date and Time Functions:- These are the functions that take values that are of datatype DATE as input and return values of datatype DATE, except for Months-between function which returns a number.

1. Sysdate():

Sysdate() is one of the data-function which is used to display the systemdate.

The function sysdate() returns a 7 byte binarydata element whose bytes represents

(i) century
(ii) year
(iii) month
(iv) day
(v) hour
(vi) minute
(vii) second.

i.e select sysdate from dual;

```
Sysdate
---------
october, 2 2017 13:28:42+0000
```

Note:-

The select statement must have a from clause. That's why oracle has a dual table. It is a special table with a column called DUMMY that has a value of 'X' used in selecting a pseudo column suchas SYS DATE.

Examples:-

① select Sysdate from dual;

```
SYSDATE
---------
26-09-17
```

Oracle enables you to extract the day, month and year from the date using an "extract" function as follows -

② select extract(day from sysdate) as only-date from dual;

```
ONLY-DATE
---------
      26
```

③ select extract(month from sysdate) as only_month from dual;

```
ONLY_MONTH
----------
      9
```

④ select extract(year from sydate) as only-year from dual;

```
ONLY_YEAR
---------
      17
```

2. Add_months(date, n):-

add_months(date, n) is also one of the date functions which is used to add specific number of months(n) to the given date. The 'n' can be both negative and positive.

Examples:-

① select add_months(sysdate, -1) as prev_month, sysdate, add-months(sysdate,1) as next_month from dual;

```
PREV-MON     SYSDATE     NEXT-MON
--------     -------     --------
26-08-17     26-09-17    26-10-17
```

② select add_moths('12 jun 2016', 12) from dual;

```
ADD-MONT
--------
24-06-17
```

③ select add-months ('24 jun 16', -12) from dual;

```
ADD-MONT
---------

24-06-15
```

④ Sample table: Reserves

SID	BID	DAY
501	101	22-06-17
502	102	23-06-17
503	103	24-06-17
504	104	25-06-17
505	105	26-06-17

Select add-moths (day, 12), sid from reserves;

```
ADD-MONT      Sid
---------     -----
22-06-18      501
23-06-18      502
24-06-18      503
25-06-18      504
26-06-18      505

5 rows selected.
```

3. **last-day (date) :-**

last-day (date) is also one of the date functions which is used to returns the last day date in the month of the specified date.

Examples :-

① select sysdate, last-day (sysdate) from dual;

```
SYSDATE       LAST-DAY
--------      --------
26-09-17      30-09-17
```

② select last_day('14 Jun 2017') from dual;

```
LAST_DAY
---------
30-06-17
```

③ Sample table: Reserves

SID	BID	DAY
501	101	22-06-17
502	102	23-06-17
503	103	24-06-17
504	104	25-06-17
505	105	26-06-17

select day, last-day(day) as last-day of day from reserves;

```
DAY         LAST-DAY OF DAY
----        ----------------
22-06-17    30-06-17
23-06-17    30-06-17
24-06-17    30-06-17
25-06-17    30-06-17
26-06-17    30-06-17
5 rows selected.
```

4. Months-between(date,date):-

Moths-between(date,date) is also one of the date functions which calculates and returns the number of months between two dates.

Examples:-

① select months-between('14 jun 2018', '14 jun 2017') from dual;

```
MONTHS-BETWEEN('14 JUN 2018', '14 JUN 2017')
--------------------------------------------
```

② select months_between('14-07-17', '24-12-12') from dual;

```
MONTHS_BETWEEN('14-07-17', '24-12-12')
--------------------------------------
                        54.6774194
```

③ Sample table: Reserves

SID	BID	DAY
501	101	22-06-17
502	102	23-06-17
503	103	24-06-17
504	104	25-06-17
505	105	26-06-17.

select sysdate, Months_between(sysdate, day), day from reserves.

```
SYSDATE      MONTHS_BETWEEN(sysdate, DAY)      DAY
--------     ----------------------------      --------
26-09-17     3.4                               22-06-17
26-09-17     3.3                               23-06-17
26-09-17     3.2                               24-06-17
26-09-17     3.1                               25-06-17
26-09-17     3                                 26-06-17.

5 rows selected.
```

5. Next_day(date, day-of-week):-

next_day(date, day-of-week) is also one of the date function which outputs the date of the first weekday specified that is later than the date.

Examples:-

① select next_day(sysdate, 'monday') as next_monday, sysdate from dual;

```
NEXT_MONDAY      SYSDATE
-----------      --------
09-10-17         02-10-17
```

sysdate,
② select × next-day(sysdate, 'friday') from dual;

```
SYSDATE        NEXT-DAY
---------      ----------
02-10-17       06-10-17
```

③ select day, next-day(day, 'friday) from reserves;

```
DAY         NEXT-DAY
----        ---------
22-06-17    23-06-17
23-06-17    30-06-17
24-06-17    30-06-17
25-06-17    30-06-17
26-06-17    30-06-17
5 rows selected.
```

Arithmetic Operations with dates and date functions :-

1. Date + number

sysdate,
select × sysdate+1 as tomarrow from dual;

```
SYSDATE       TOMARROW
--------      ---------
02-10-17      03-10-17
```

2. Date - number

select sysdate, sysdate+1 as yesterday from dual;

```
SYSDATE       YESTERDAY
--------      ----------
02-10-17      01-10-17.
```

3. Date - Date

sysdate, lastday
~~sys~~
select × last-day(sysdate) - sysdate as days-left, × from dual;

```
DAYS-LEFT      SYSDATE      LASTDAY
---------      --------     --------
                            31-10-17
30             02-10-17
```

This command displays, how many days are left in the current month except current day.

Numeric-functions :-

There are functions that accept a numeric input and return numeric values. Numeric functions are used to perform operations on numbers.

1. ABS(x) :-

ABS(x) means absolute value of the number 'x'. means it returns the absolute value of the given number x. It accepts both +ve & -ve values

Examples:-

① select abs(1) from dual;

```
ABS(1)
-----------
          1
```

② select abs(-1) from dual;

```
  ABS(-1)
------------
          1
```

2. CEIL(x) :-

CEIL(x) means integer value that is greaterthan or equals to the number x. It accepts both +ve and -ve values.

Examples:-

① select ceil(8.235) from dual;

```
  CEIL(8.235)
-------------
            9
```

② select ceil(-29.33) from dual;

```
  CEIL(-29.33)
--------------
          -29
```

3. FLOOR(x):-

FLOOR(x) is a numeric function that returns integer value that is less than or equal to the number 'x'. It accepts both +ve and -ve values.

Examples:-

① select floor(8.235) from dual;

```
FLOOR(8.235)
-------------
          8
```

② select floor(-29.33) from dual;

```
FLOOR(-29.33)
-------------
         -30
```

③ select floor(30) from dual;

```
FLOOR(30)
----------
       30
```

4. TRUNC(X,Y):-

Trunc(x,y) is a numeric function that truncates value of numbe 'x' upto 'y' decimal places. Trunc() function can have one argument or two arguments.

Examples:-

① select trunc(8.639) from dual;

```
TRUNC(8.639)
------------
      8
```

② select trunc(8.6391345, 2) from dual;

```
TRUNC(8.6391345,2)
------------------
         8.63
```

③ select trunc(bid) from reserves;

```
TRUNC(BID)
----------
       101
       102
       103
       104
       105
 5 rows selected.
```

5. ROUND(X,Y):-

Round(x,y) is a numeric function that rounded off value of the number 'x' to the number 'y' decimal places.

Examples:-

① select round(86.546321) from dual;

```
ROUND(86.546321)
----------------
              87
```

② select round(86.546321,2) from dual;

```
ROUND(86.546321,2)
------------------
             86.55
```

③ select round(bid) from reserves;

```
ROUND(bid)
----------
       101
       102
       103
       104
       105.
 5 rows selected.
```

String Functions :-

string functions are also known as Character or Text functions. Character or Text functions are used to manipulate text strings. They accept characters or strings as input and can return both character/strings and numerical values as output.

1. lower(string-value) :-

lower(string-value) is one of the string functions which converts and returns all the letters in string-value in lower case.

Example:-

① select lower('HARIKA') from dual;

```
LOWER('HARIKA')
---------------
harika
```

② Sample table : Sailors

SNAME	SID
RAVI	501
RAJU	502
RAM	503
RAJESH	504

select lower(sname) from sailors;

```
LOWER(SNAME)
--------------------
ravi
raju
ram
rajesh

4 rows selected.
```

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

2. UPPER(String-value) :-

Upper(string-value) is also one of the string functions that converts all the letters in string-value to Capital letters.

Examples

① select upper('harika') from dual;

```
UPPER('harika')
---------------
HARIKA
```

② select upper(sname) from sailors;

```
UPPER(SNAME)
------------
RAVI
RAJU
RAM
RAJESH.
```

~~③ Tr~~ 4 rows selected.

3. initcap(string-value) :-

initcap(string-value) is also one of the string functions that converts all the letters in string-value to mixed case(i.e initial letter capital & remaining are smalls).

Examples

① select initcap('harika') from dual;

```
INITCAP('harika')
-----------------
Harika
```

② select initcap(sname) from sailors;

```
INITCAP(SNAME)
--------------
Ravi
Raju
Ram
Rajesh.
```

4 rows selected.

4. Ltrim(string-value, 'trim-text'):-

Ltrim(string-value,'trim-text') f is also one of the string functions which trims/removes all the occurrences of 'trim-text' from the left of the 'string-value'.

Examples

① select ltrim('harika','h') from dual;

```
LTRIM
-----
arika
```

② select sname, ltrim(sname,'r') from sailors;

```
SNAME     LTRIM(SNAME,'R')
-----     ----------------
RAVI      AVI
RAJU      AJU
RAM       AM
RAJESH    AJESH
4 rows selected.
```

5. Rtrim(string-value,'trim_text'):-

Rtrim(string-value,'trim-text') is also one of the string functions which trims/removes all the occurrences of 'trim-text' from the right of the 'string-value'.

Examples

① select rtrim('harika', 'a') from dual;

```
RTRIM(
------
harik
```

② select sname, rtrim(sname,'h') from sailors;

```
SNAME     RTRIM(SNAME,'H')
-----     ----------------
Ravi      Ravi
Raju      Raju
Ram       Ram
Rajesh    Rajes

4 rows selected.
```

6. Trim(string-value, 'string-text') :-

Trim(string-value,'string-text') is also one of the string functions which trims/ removes the 'string-text' from both left and right of the 'string-value'.

Examples

① select trim('Pradeep', 'p') from dual;

```
TRIM(
-------
radee
```

② select sname, trim(sname,'r') from sailors;

```
SNAME   TRIM(SNAME,'R')
-----   ---------------
Ravi      avi
Raju      aju
Ram       am
Rajesh    ajesh

4 rows selected.
```

7. Substr(string-value, m,n) :-

substr(string-value, m,n) is also one of the string functions which returns 'n' number of characters from string-value starting from the 'm' position.

Examples :-

① select substr('harika', 2, 4) from dual;

```
SUBSTR
-------
arik
```

② select substr('harika', 2) from dual;

```
SUBSTR
-------
arika
```

③ select sname, substr(sname, 2, 2) from sailors;

```
SNAME    SUBSTR(SNAME,2,2)
-----    -----------------
Ravi       av
Raju       aj
Ram        am
Rajesh     aj
```

④ select sname, substr(sname,2) from sailors;

```
SNAME        SUBSTR(SNAME,2)
------       ---------------
Ravi         avi
Raju         aju
Ram          am
Rajesh       ajesh

4 rows selected.
```

8. Length(string-value) :-

length(string-value) is also one of the string functions which returns the number of characters in string-value.

Examples

① select ~~harika~~ length('harika') from dual;

```
LENGTH('HARIKA')
----------------
               6
```

② select sname, length(same) from sailors;

```
SNAME        LENGTH(SNAME)
------       -------------
Ravi             4
Raju             4
Ram              3
Rajesh           6

4 rows selected.
```

9. Lpad(string-value, n, pad-value) :-

Lpad(string-value, n, pad-value) is also one of the string function that returns string-value left padded with 'pad-value'. The length of the whole string will be of 'n' characters.

Tutorials Duniya.com @codes.learning

Examples

① select lpad('harika', 15, '*') from dual;

```
LPAD('HARIKA', 15, '*')
-----------------------
*********harika
```

② select lpad(sname, 10, '*') , ~~from~~ sname from sailors;

```
LPAD(SNAME,10,'*')      SNAME
------------------      -----
      ******Ravi        Ravi
      ******Raju        Raju
      *******Ram        Ram
      ****Rajesh        Rajesh

4 rows selected.
```

10. **Rpad(string-value, n, Pad-text) :-**

Rpad(string-value, n, pad-text) is also one of the string functions that returns string-valuee right-padded with pad-text. The length of the whole string will be of 'n' characters.

Examples

① select rpad('harika', 15, '*') from dual;

```
RPAD('HARIKA', 15, '*')
-----------------------
harika*********
```

② select ~~rpad(~~ sname, rpad(same, 10, '*') from sailors;

```
SNAME      RPAD(SNAME,10,'*')
-----      ------------------
Ravi       Ravi******
Raju       Raju******
Ram        Ram*******
Rajesh     Rajesh****

4 rows selected.
```

11· Concate(string-value1, string_value2) :-

Concat(string-value1, string-value2) is also one of the string function used to Concatenation purpose. i.e it concates string-value1 and string-value2 .

Examples

① select concat('harika', 'ravuri') from dual;

```
CONCAT('HARIKA', 'RAVURI')
-------------------------

harikaravuri
```

② select sname, sid, Concat(sname, sid) from sailors;

```
SNAME     SID     CONCAT(SNAM, SID)
-----     ----    ----------------
ravi      501     ravi501
raju      502     raju502
ram       503     ram503
rajesh    504     rajesh504

4 rows selected.
```

12· Instr(string-value, symbol) :-

instr(string-value, symbol) is also one of the string function which return the first index value of the symbol in the string-value.

Examples :-

① select instr('harika', 'a') from dual;

```
INSTR('HARIKA', 'A')
--------------------
                   2
```

② select instr(Sname, 'j'), sname from sailors;

```
INSTR(SNAME, 'J')    SNAME
-----------------    -----
        0            ravi
        3            raju
        0            ram
        3            rajesh
```

https://www.tutorialsduniya.com Scanned by CamScanner

Conversion functions :-

These are the functions that help us to convert a value in one form to another form. For example, a null value into on actual value, or a value from one datatype to another datatype.

1. To_char() :-

to_char() function converts a number or date to a string & returns a string.

Syntax :-

```
to_char(value[, format_mask] [, nls_language])
```

where

value → a number or date that will be converted to a string.

format_mask → optional. This is the format that will be used to convert a value to a string.

nls_language → optional. This is the nls language used to convert value to a string.

Examples with numbers :-

① select to_char(1210.73, '9999.9') from dual;

```
TO_CHAR(1210.73, '9999.9')
--------------------------
          '1210.7'
```

② select to_char (1210.73, '9,9999.99') from dual;

```
TO_CHAR(1210.73,'9,999.99')
---------------------------
        '1,210.73'
```

③ select to_char (1210.73, '$9,999.00') from dual;

```
TO_CHAR(1210.73,'$9,999.00')
----------------------------
        '$1,210.73'
```

④ ~~test~~ select to-char(21,'0000091') from dual;

```
TO-CHAR(21,'0000091')
----------------------
      '0000021'
```

Examples with Dates :-

The following is a list of valid parameters when to-char function is used to convert a date to a string. These parameters can be used in many combinations.

Parameter	Explanation
YEAR	Year, spelledout
yyyy	4 digit year
yyy	Last 3, 2, or 1 digit(s) of ISO year.
yy	
y	
IYYY	4 digit year based on the ISO standard.
MM	Month((01-12) JAN=01)
MON	abbreviated name of Month.
MONTH	Name of month, padded with blanks to length of 9 characters.
RM	Roman numeral month (I- XII, JAN=I)
WW	Week of year (1-53).
W	Week of month (1-5).
IW	Week of year (1-52 or 1-53) based on ISO standard.
D	Day of week (1-7)
DAY	Name of day.

DD	Day of moth (1-31)
DDD	Day of year (1-366)
DY	abbreviated name of day.
HH	hour of day (1-12)
HH12	hour of day (1-12)
HH24	hour of day (0-23)
MI	Minute (0-59)
SS	second (0-59)
SSSS	Seconds past midnight (0-86399)
FF	fractional seconds.

①
```
select to_char(sysdate, 'yyyy/MM/DD') ,sysdate from dual;

TO_CHAR( SYSDATE,' yyyy/MM/DD)   SYSDATE
------------------------------   --------
      2017/09/02.                02-09-17
```

②
```
select sysdate, to_char(sysdate, 'MONTH DD, YYYY') from dual;

SYSDATE   TO_CHAR(SYSDATE,'MONTH DD, YYYY')
-------   ---------------------------------
02-09-17      September 02, 2017.
```

③
```
select sysdate, to-char(sysdate, 'FMMonth DD, YYYY') from dual;

SYSDATE    TO_CHAR(SYSDATE,'FMMONTH DD,YYYY')
-------    ----------------------------------
02-09-17     september  2,2017.
```

FM- used to supress the zero's and blanks.

④
```
select sysdate, to-char(sysdate, 'MON DDth, YYYY') from dual;

SYSDATE     TO-CHAR(SYSDATE,'MON DDth,YYYY')
--------    --------------------------------
02-09-17      Sep 02nd, 2017
```

⑤ sample table : Reserves

SID	DAY
101	22-06-17
102	22-07-17
103	22-08-17
104	22-09-17

Select day, to_char(day, 'MON DD, YYYY') from reserves;

DAY	TO_CHAR(day, 'MON DD, YYYY')
22-06-17	JUN 22, 2017
22-07-17	JUL 22, 2017
22-08-17	AUG 22, 2017
22-09-17	SEP 22, 2017

⑥ Number to string conversion Example

Sample table: Reserves.

SID	DAY
101	22-06-17
102	22-07-17
103	22-08-17
104	22-09-17

Select sid, to_char(sid, '$000999') from reserves;

SID	TO_CHAR(SID, '$000999')
101	$000101
102	$000102
103	$000103
104	$000104

2. To-date():-

To_date() function is used to convert a string to a date and returns a date value as output.

Examples:-

① ~~To~~ select to-date('2003/07/09', 'yyyy/mm/dd') from dual;

```
TO_DATE('2003/07/09','YYYY/MM/DD')
----------------------------------
July 9, 2003.
```

② select to-date('070903', 'MMDDYY') from dual;

```
TO-DATE('070903','MMDDYY')
--------------------------
July 9, 2003
```

③ select to-date('20020315', 'yyyyMMDD') from dual;

```
TO-DATE('20020315','YYYYMMDD')
------------------------------
Mar 15, 2002.
```

3. To_number():-

To-number() function is used to convert a string to a number and returns a number value as output.

Examples:-

① select to-number('1210.73') from dual;

```
TO_NUMBER('1210.73')
--------------------
1210.73
```

② select to-number('526') from dual;

```
TO-NUMBER('526')
----------------
526.
```

https://www.tutorialsduniya.com Scanned by CamScanner

Overview

- There are two types of functions (built-in) in oracle.

  1. Single row / scalar functions
  2. Group functions.

- Single row / scalar functions are –
  - (i) Numeric functions
  - (ii) String functions
  - (iii) date and time function
  - (iv) conversion functions.

- Group functions are –
  - (i) Aggregate functions.

- Numeric functions are –
  - a) abs(x)
  - b) ceil (x)
  - c) floor(x)
  - d) Trunc().
  - e) Round()

- String functions are –
  - a) lower()
  - b) upper()
  - c) initcap()
  - d) ltrim()
  - e) rtrim()
  - f) trim()
  - g) lpad()
  - h) rpad()
  - i) substr()

j) instr( )

k) length( )

l) concat ( )

- Date functions are –

  a) add-months( )

  b) months-between( )

  c) next-day( )

  d) last-day( )

  e) sysdate( )

  some other date functions are –

  f) least( )

  g) greatest( )

  h) trunc( )

  i) round ( )

- Conversion functions are –

  a) to-char( )

  b) to-date( )

  c) to-number( )

- Aggregate functions are –

  a) sum( )

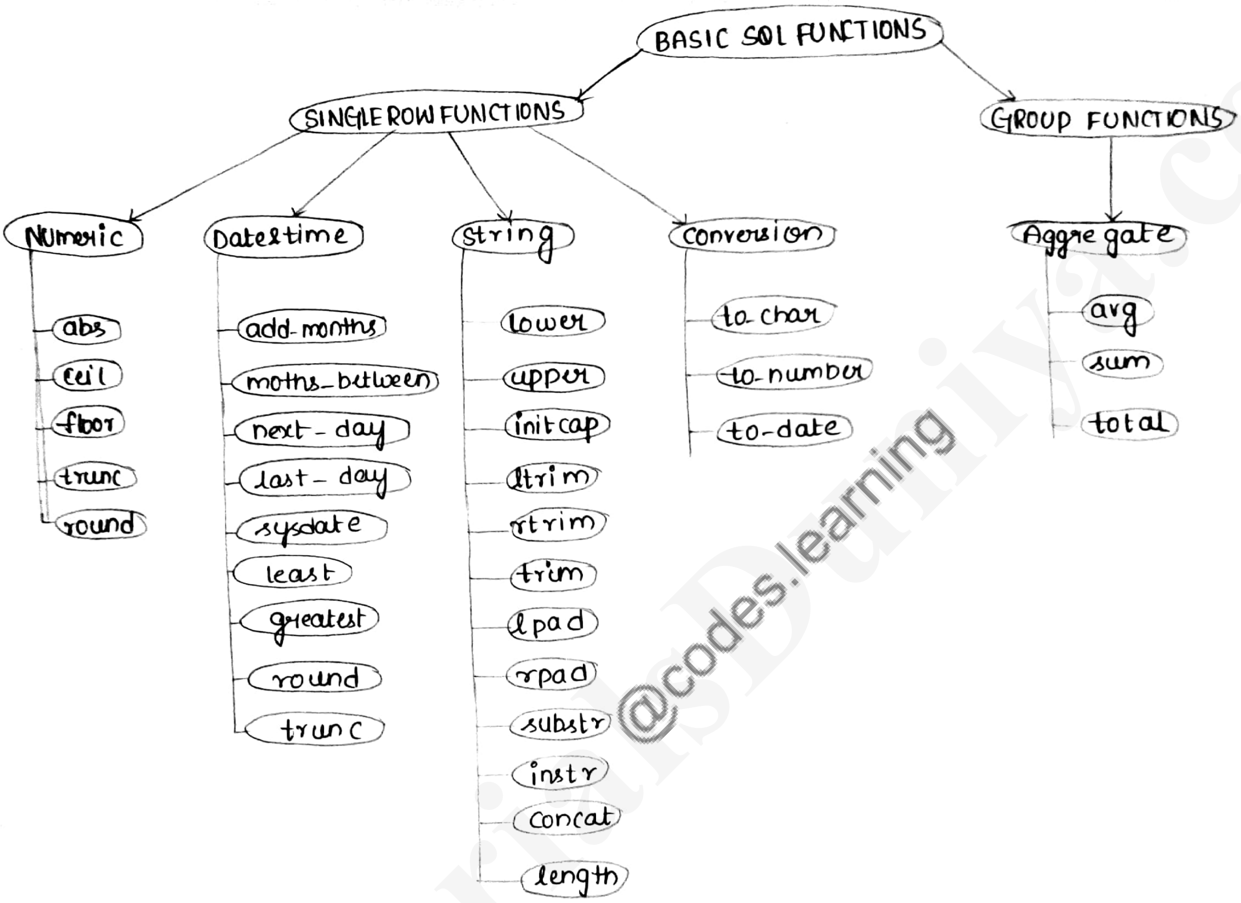
  b) avg ( )

  c) total( ) etc.

- Dual table :–

This is a single row and single column dummy table provided by DB. This is used to perform mathematical calculations without using table. select * from dual;

```
DUMMY
-------
X
```


BASIC SQL FUNCTIONS
SINGLE ROW FUNCTIONS
GROUP FUNCTIONS
Numeric
abs
ceil
floor
trunc
round
Date&time
add-months
moths_between
next-day
last-day
sysdate
least
greatest
round
trunc
String
lower
upper
initcap
ltrim
rtrim
trim
lpad
rpad
substr
instr
concat
length
Conversion
to-char
to-number
to-date
Aggregate
avg
sum
total

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- **Algorithms Notes**
- **Artificial Intelligence**
- **Android Programming**
- **C & C++ Programming**
- **Combinatorial Optimization**
- **Computer Graphics**
- **Computer Networks**
- **Computer System Architecture**
- **DBMS & SQL Notes**
- **Data Analysis & Visualization**
- **Data Mining**
- **Data Science**
- **Data Structures**
- **Deep Learning**
- **Digital Image Processing**
- **Discrete Mathematics**
- **Information Security**
- **Internet Technologies**
- **Java Programming**
- **JavaScript & jQuery**
- **Machine Learning**
- **Microprocessor**
- **Operating System**
- **Operational Research**
- **PHP Notes**
- **Python Programming**
- **R Programming**
- **Software Engineering**
- **System Programming**
- **Theory of Computation**
- **Unix Network Programming**
- **Web Design & Development**

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

@codes.learning

Explain in detail about cluster and Multilevel indexes.

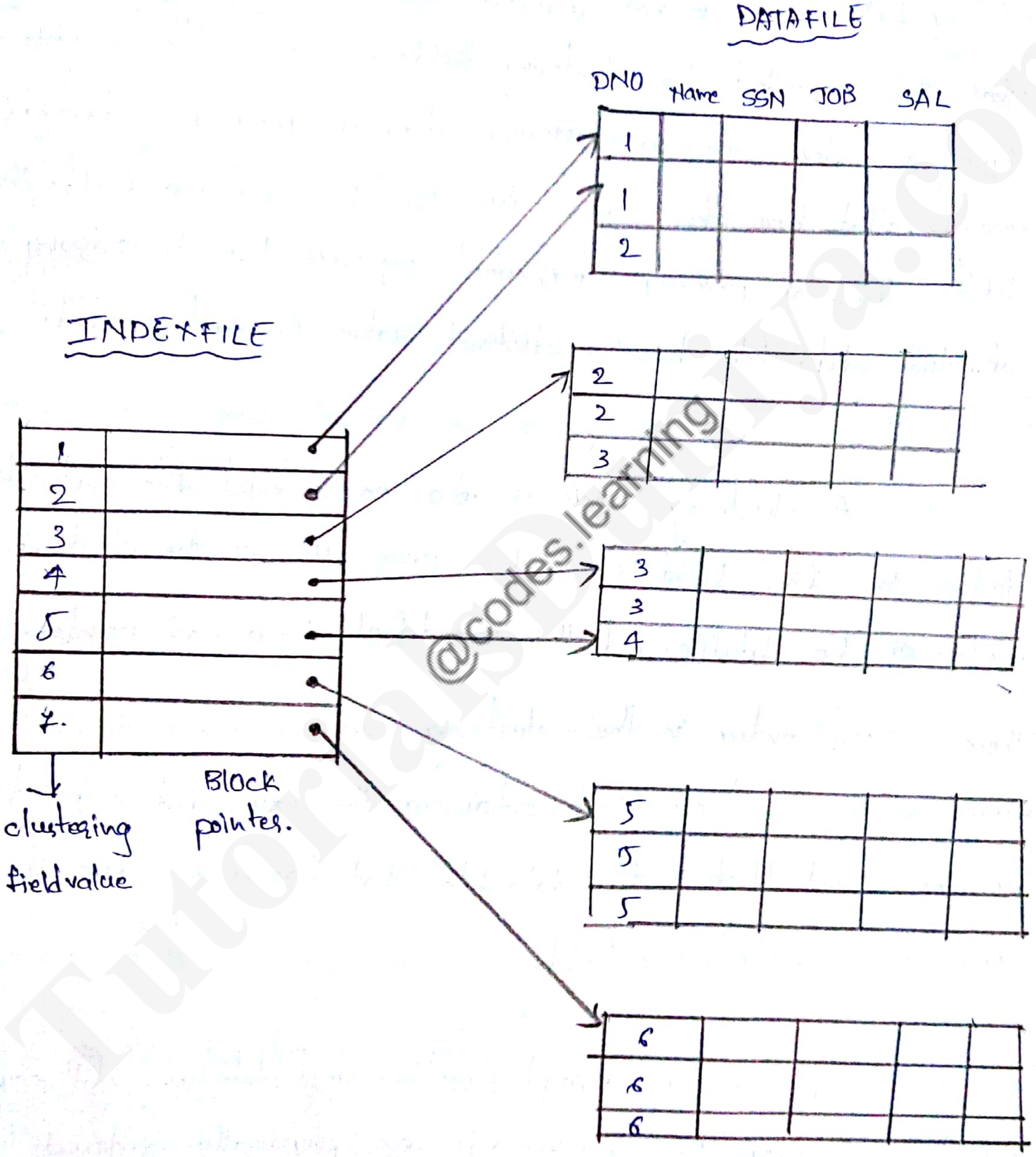
Clustering indexes :-

If records of a file are physically ordered on a nonkey field - which doesnot have a distinct value for each record. that field is called the clustering field. We can create a different type of index, called a clustering index, to speed up retrieval of records that have the same value for the clustering field. This differs from a primary index, which requires that the ordering field of the data file have a distinct value for each record.

A clustering index is also an ordered file with two fields, the first field is of the same type as the clustering field. of the datafile. and the second field is a block pointer. There is one entry in the clustering index for each distinct value of the clustering field, containing the value and a pointer to the first block in the data file that has a record with that value for its clustering field.

Notice that record insertion and deletion still cause problems, because the data records are physically ordered. To alleviate the problem of each insertion, it is common to reserve a whole block (or a cluster of contiguous blocks) for each value of the clustering field. all records with that value are placed in the block. This makes insertion and deletion relatively straight forward.

A clustering index is another example of a nondense index, because it has an entry for every distinct value of the indexing field rather than for every record in the file.


DATAFILE
DNO
Name
SSN
JOB
SAL
1
1
2
2
2
3
3
3
4
5
5
5
6
6
6
INDEXFILE
1
2
3
4
5
6
7.
clustering fieldvalue
Block pointer.

Multilevel indexes:-

The idea behind a multilevel index is to reduce the part of the index that we continue to search. by bfri, the blocking factor for the index, which is larger than 2. Hence, the search space is reduced much faster. The value bfri, is called the fan-out of the multilevel index, and we will refer to it by the symbol fo. Searching a multilevel index requires approximately ($\log_{fo} b_i$) block. access which is a smaller number than for binary search if the fanout is larger than 2.

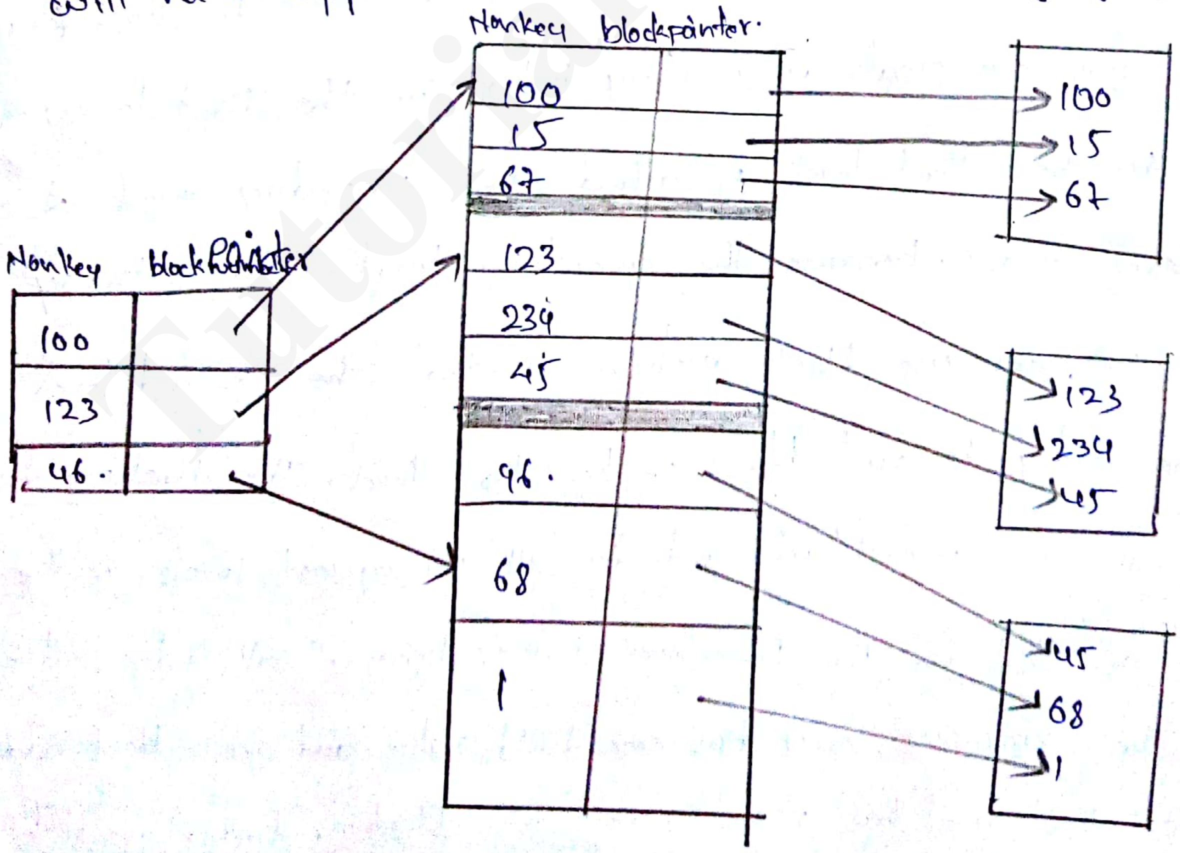
A multilevel index considers the index file, which we will now refer to as the first (or base) level of a multilevel index, as an ordered file with a distinct value for each K(i). Hence we can create a primary index for the first level; this index to the first level is called the secondary level of the multilevel index. Because the secondary level is a primary index, we can use block anchors so that the second level has one entry for each block of the first level. The blocking factor bfri of the second level - and for all subsequent levels - is the same as that for the first-level index, because all index entries are the same size; each has one field value and one block address. if the first level has r1 entries and the blocking factor -which is

tutorialsduniya.com @codes.learning

https://www.tutorialsduniya.com Scanned by CamScanner

also the fan-out for the index is $bfr_i = f_o$; then the first level needs (r_1/f_o) blocks, which is therefore the number of entries r_2 needed at the second level of the index.

We can repeat this process for the second level. The third level, which is a primary index for the second level. has an entry for each second-level, so the number of third-level entries is $r_3 = (r_2/f_o)$. we can repeat this process untill all the entries of some index level t fit in a single block. this block at the t th level is called the top index level. Each level reduces the number of entries at the previous level by a factor of f_o - the index fan-out - so we can use the formula $1 \leq (r_1/((f_o)^t))$ to calculate t. Hence a multilevel index with r_1 first-level entries will have approximately t levels. where $t = (\log_{f_o}(r_1))$.


Nonkey blockpointer
100
123
46
Nonkey blockpointer
100
15
67
123
234
45
96
68
1
100
15
67
123
234
45
45
68
1

By Considering an Example, show how to reduce access time with primary index.

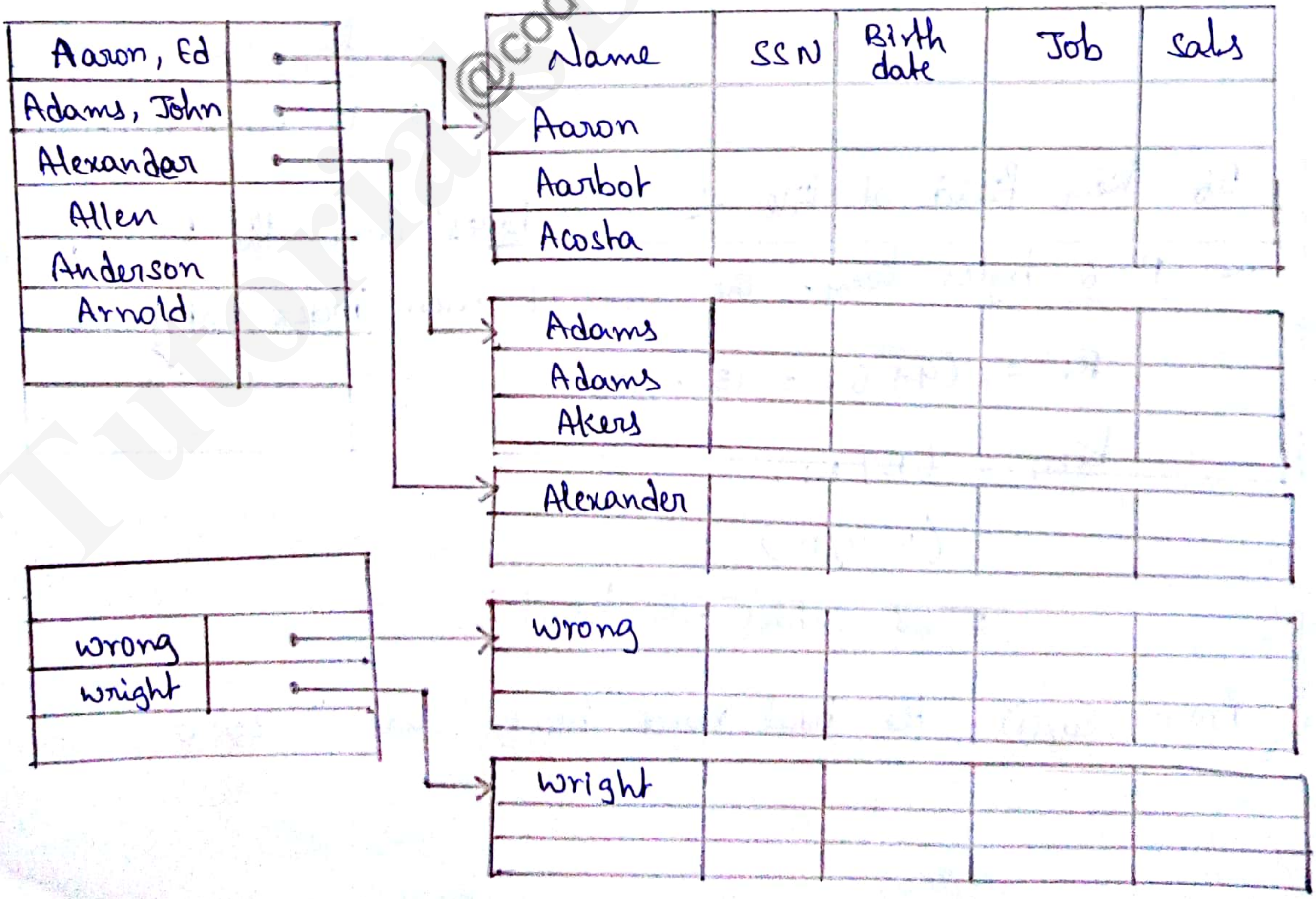
- Primary index :- It is an ordered file whose records are of fixed length with 2 fields. The first field is of the same datatype as ordering key field called primary key of datafile. Second field is a pointer to disk block. we will refer these 2 field values of index Entry i as <k(i),P(i)>

To create a primary index on the ordered file is shown in fig. we use NAME field as primary key.

< k(1) = (Aaron, Ed), P(1) = address of block1 >

< k(2) = (Adams, John), P(2) = address of block2 >

< k(3) = (Alexander, Ed), P(3) = address of block3 >


Aaron, Ed
Adams, John
Alexander
Allen
Anderson
Arnold
wrong
wright
Name
SSN
Birth date
Job
Sals
Aaron
Aarbot
Acosta
Adams
Adams
Akers
Alexander
wrong
wright

Ex:- Illustrate the saving in block accesses that is attainable when a primary index is used to search for a record.

→ Suppose that we have an ordered file that $r = 30,000$ records stored on a disk with blocksize $B = 1024$ bytes.

File records are of fixed size & are with record length $R = 100$ bytes.

The blocking factor for file would be $bfr = (B/R)$

$= (1024/100)$

$= 10$ records per block.

→ No. of blocks needed for the file is $b = (r/bfr)$

$= (30,000/10)$

$= 3000$ blocks $\longrightarrow$ ①

→ Binary search the data file would be $\log_2 b$.

$= \log_2 3000$

$= 12$ block accesses $\longrightarrow$ ②

If key field of file is $V = 9$ bytes long the block pointer is $P = 6$ bytes long. The size of each index entry is

$R_i = (9+6) = 15$ bytes.

$bfr_i = (B/R_i)$

$= (1024/15)$

$= 68$ Entries per block

From Eq ①, the total no. of blocks are " 3000 "

The no. of index blocks is $b_i = (r_i / bfr_i)$

$= (3000/68)$

$= 45$ blocks $\longrightarrow$ ③

$(\log_2 b_i) = (\log_2 45)$

$= 6$ block access $\longrightarrow$ ④

we need one additional block to data file for total of $6+1=7$ block access.

→ From eq ①, we have 3000 blocks, it access 12 blocks.

→ But in eq ③, we access the 45 blocks and time access only 6 block. So, it reduces access time. A major problem with a primary index. as with any ordered file is insertion & ~~de~~ deletion of records.

Q Distinguish b/w Ordered indexing and Hashing.

A:- Each scheme has advantages in certain situations. And the DBMS implementor could leave the decision to the database designer and provide several methods. Normally, the implementor only provides a very limited number of schemes.

Typically, ordered indexing is used unless it is known in advance that range queries will be in frequent, in which case hashing is used. Hash organizations are particularly useful for temporary files created using query processing if lookup on a key value are required and no ranges queries will be performed.

Index Definition in SQL

creating an index:

```
create Index <index-name> ON
   <relation-name> (<attribute_list>)
```

Example:

```
CREATE INDEX branch_Index ON
branch (branch_name)
```

Deleting an Index:

```
DROP INDEX <index-name>
```

Comparision of Indexing and Hashing

(1) To make a wise choice b/w the methods to seek, database designer must consider the following issues.

✓ Is the cost of periodic re-organization of Index or hash structure acceptable?

✓ what is the relative frequence of insertion and deletion?

✓ Is it desirable to optimize average access time at the expense of increasing worst-case access time?

✓ what types of queries are users likely to pose?

(2) The last issue is critical to the choice b/w indexing and hashing. If most queries are of the form. then to process this query the system will perform a lookup on an index or hash structure for attribute with value c.

(3) For these sort of queries hashing scheme is preferable

✓ Index lookup takes the proportional to log of number of values in R for.

✓ Hash structure provides look up average time that is a small constant (independent of database size)

(4) However the worst case follows indexing:

✓ Hash worst case gives time proportional to the number of value in R for.

✓ Index worst case still log of number of values in R

(5) Index methods are preferable where a range of values in specified in the query eg.

This query finds records with values in the range from to

@codeLearning

(6) using an index structure, we can find the bucket for value, and then follow the pointer chain to read the next buckets in alphabetic (or numeric) order until we find.

✓ If we have a hash structure instead of an index we can find a bucket for easily, but it is not easy to find the "next bucket".

✓ A good hash function assigns values randomly to buckets

✓ also, each bucket may be assigned many search key values, so we cannot chain them together.

✓ To support range queries using a hash structure, we need a hash function that preserves order.

✓ For example, if k_1 and k_2 are search key values and $k_1 < k_2$ then $h(k_1) < h(k_2)$

✓ Such a function could ensure that also provide randomness and uniformity are extremely difficult to find. and they are in key order.

✓ Order-preserving hash functions that also provide randomness and uniformity are extremely difficult to find.

✓ Thus most systems use indexing in preference to hashing unless it is known in advance that range queries will be infrequent

Data Base Management Systems

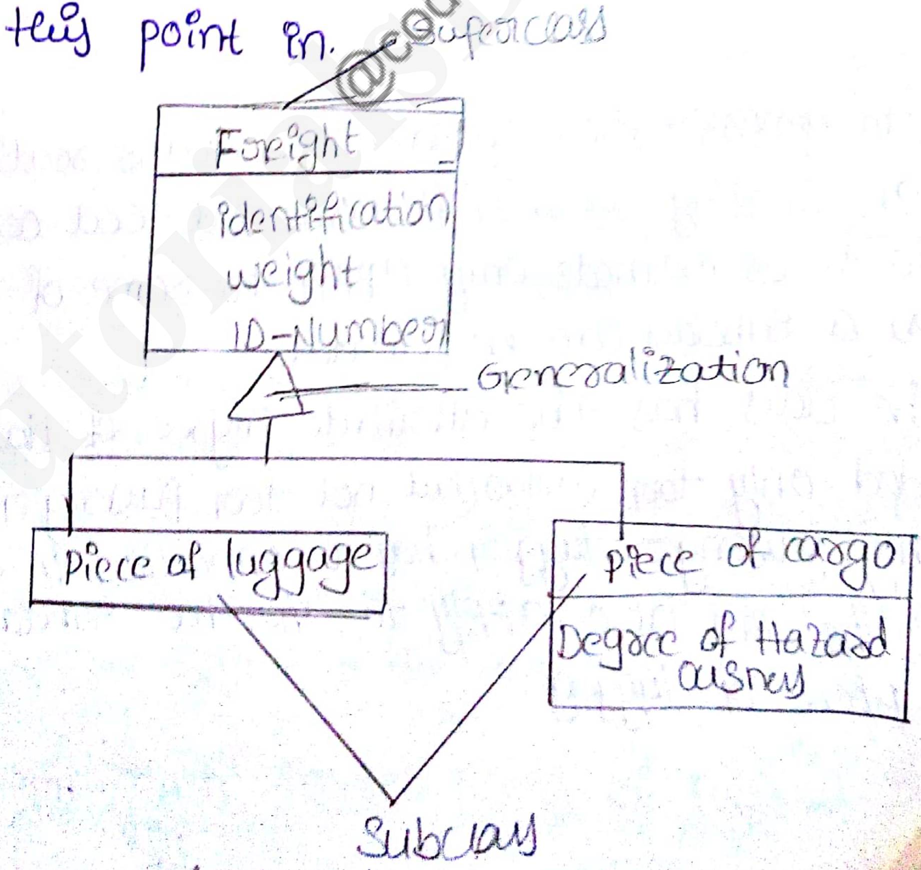
R. Munisha
15F01A05E1
CSE.C

3. E-R MOdels

Discuss the mechanism of attribute relationship Inheritance. How is it useful?

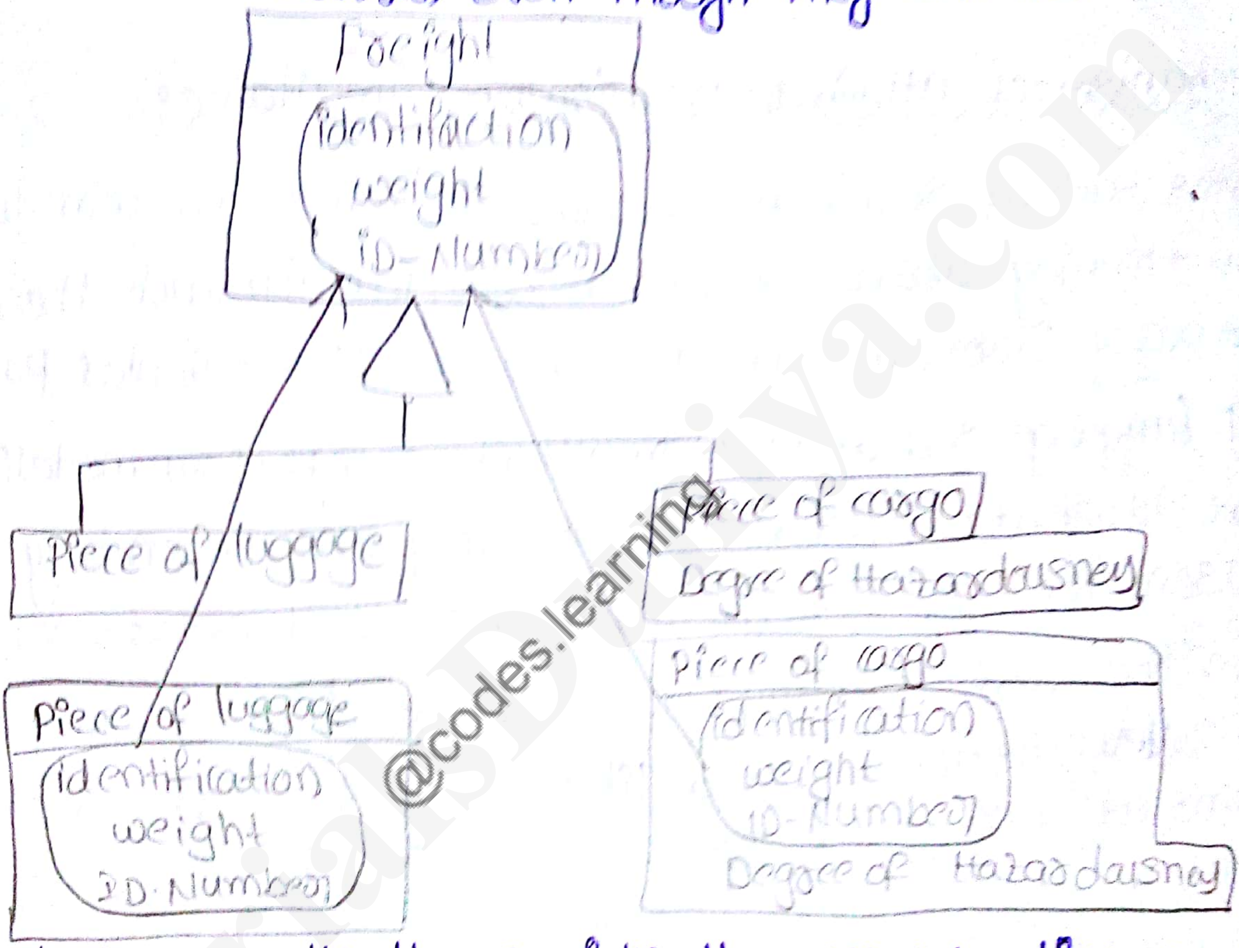
mechanism of attribute relationship Inheritance:-

Terms such as superclass, subclass, or inheritance come to mind when thinking about the object-oriented approach. these concepts are very important when dealing with object oriented programming language such as java, smalltalk, or c++. For modeling classes that illustrate techinal concepts they are secondary. the reason for this is that modeling relevant objects or ideas from the real world gives little opportunity for using inheritance Nevertheless, we would like to further introduce these terms at this point in.


Superclass
Freight
identification
weight
ID-Number
Generalization
Piece of luggage
piece of cargo
Degree of Hazard
ausney
Subclass

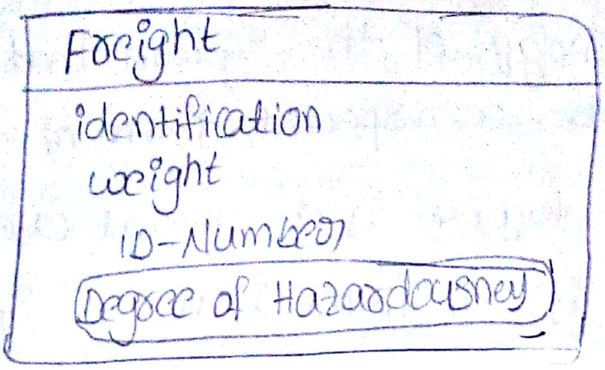
Generalization is the process of Extracting shared characteristics from two or more classes, and combining them into a generalized super-class. Shared characteristics can be "attributes", "associations," or methods.

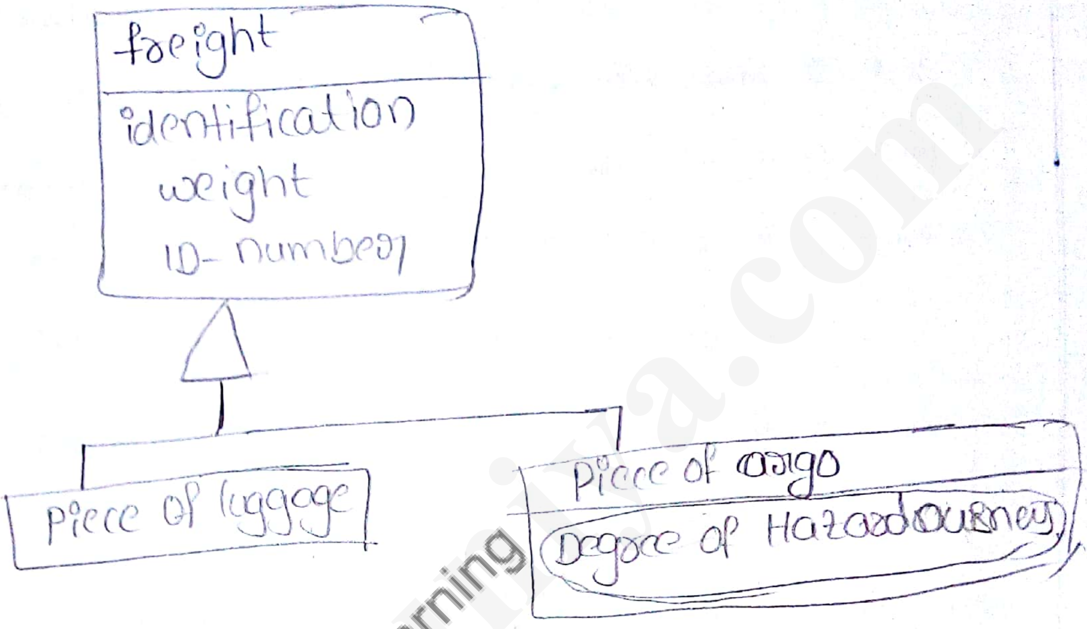
The shared attributes are only listed in the superclass but also apply to the two subclasses, Even though they are not listed there


Freight
identification
weight
ID-Number
Piece of luggage
Piece of cargo
Degree of Hazardousness
Piece of luggage
identification
weight
ID-Number
Piece of cargo
identification
weight
ID-Number
Degree of Hazardousness

In contrast to generalization, specialization means creating new sub-classes from an existing class, If it turns out that certain attributes, associations, or methods only apply to some of the objects of the class, a subclass can be created.

In the class has the attribute Degree of Hazardousness which is needed only for cargo, but not for passenger luggage. Additionally only passenger lugger. has a connection to a coupon the attributes of the class also apply to the two subclasses. piece of cargo and piece of luggage.


Freight
identification
weight
ID-Number
Degree of Hazardousness


freight
identification
weight
ID-number
Piece of luggage
Piece of cargo
Degree of Hazardousness

So much for the mechanism. However, the domain meaning of the relationship between superclass and subclass is much important. these rules apply to this relationship.

* All statements that are made about a super class. also to all subclasses. we say that subclasses "inherit" attributes, associations, and operations from the superclass.

Ex:- If the superclass Freight has an attribute weight, then the subclass piece of luggage also has an attribute weight, even though this attribute is not listed in the subclass Piece of luggage.

* Anything that can be done with an object of the superclass can be done with an object of the subclass.

Ex:- If freight can be loaded, pieces of luggage can be loaded.

*In the terminolgy of the System that is being modeled, a subclass has to be a special form of the superclass.

ex:- A piece of luggage is a special case of freight the counte
Example to this is: A flight is not a special case of a flight number.

Q: By considering an example describe various data update operations in SQL.

A: An SQL update statement changes the data of one or more records in a table. Either all the rows can be updated, or subset may be chosen using a condition.

The update statement has the following form

```
UPDATE table_name SET column_name = value [, column_name
                                  = value ---] [WHERE condition]
```

For the update to be succesful, the user must have data manipulation previliges (update previlige) on the table or column and the updated value must not conflict with all the applicable constraints (such as primary keys, unique indexes, CHECK constraints, and NOTNULL constraints).

Examples:

① Set the value of column c1 in table T to 1, only in those rows where the value of column c2 is "a".

```
UPDATE T
    SET C1 = 1
  WHERE C2 = 'a'
```

② In a table T, set the value of column C1 to 9 and the value of C3 to 4 for all rows for which the value of column C2 is "a".

```
UPDATE T
   SET C1 =9,
       C3 =4.
   WHERE C2='a'
```

③ Increase value of column C1 by 1 if value in column C2 is 'a'

```
UPDATE T
   SET C1 = C1+1
   WHERE C2 ='a'
```

④ Prepend the value in column C1 with the string "text" if the value in column C2 is "a".

```
UPDATE T
   SET C1 = 'text' || C1
     WHERE C2 = 'a'.
```

⑤ Set the value of column C1 in table T1 to 2, only if the value of column C2 is found in the sublist of values in column C3 in table T2 having column C4 equal to 0.

```
UPDATE T1
    SET C1 =2
   WHERE C2 IN (SELECT C3
                 FROM T2
                 WHERE C4 =0)
```

⑥ One may also update multiple columns in a single update statement.

```
UPDATE T
    SET C1 = 1,
        C2 = 2
```

⑦ Complex conditions and JOINS are also possible.

```
UPDATE T
    SET A = 1
  WHERE C1 = 1
    AND C2 = 2
```

⑧ Some databases allow the non-standard use of FROM clause.

```
UPDATE a
   SET a.[updated_column] = updatevalue
  FROM articles a
       JOIN classification a
         ON a.articleID = c.articleID
       WHERE c.classID = 1
```

⑨ Or on oracle systems (assuming there is an index on classification.articleID).

```
UPDATE
(  SELECT *
     FROM articles
     JOIN classification
       ON articles.articleID = classification.articleID
    WHERE classification.classID = 1
)
SET [updated_column] = update value.
```

Q:- Differentiate between Specialization & generalization.

A) Class Hierarchies (Specialization & generalization)

Specialization	generalization
1. process of identifying subsets of entity set that share some special distinguishable characteristics.	1. Process of identifying some generalized (common) entity set and creating a new entity set that contain these common features.
2. Here superclass is defined first then subclasses are defined next.	2. Here subclasses are defined first, then super class is defined next.
3. It is top-down process. ie; super class is divided into subclasses based on distinguishable features.	3. It is bottom-up process. ie; subclasses are synthesized into superclass based on common features.
4. Speciality can be expressed by specialization. Eg:- 'employees' is specialized into 'permenant', & Temperory.	4. Commonality can be expressed by generalization. Eg:- permenant & Temperory employees are generalized by employees.

→ Specialization is the inverse process of generalization.

→ Subclasses (subsets) can be either mutually exclusive category (or) overlapping. An example of mutually exclusive category would be an employee who works in an organization the employee can either be permanent or temperory but not both.

→ Overlapping category is when an entity may be in two or more subclasses. An example would be a staff member who works in a university and could also be a student of the same university.

→ The subclass entities can also be total (or) partial based (on) their participation in relationship.

→ Class hierarchies can be nested. ie, A sub-class of hierarchy can be a super class of another.

→ The generalization / Specialization is represented with a triangle labeled as ISA.

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

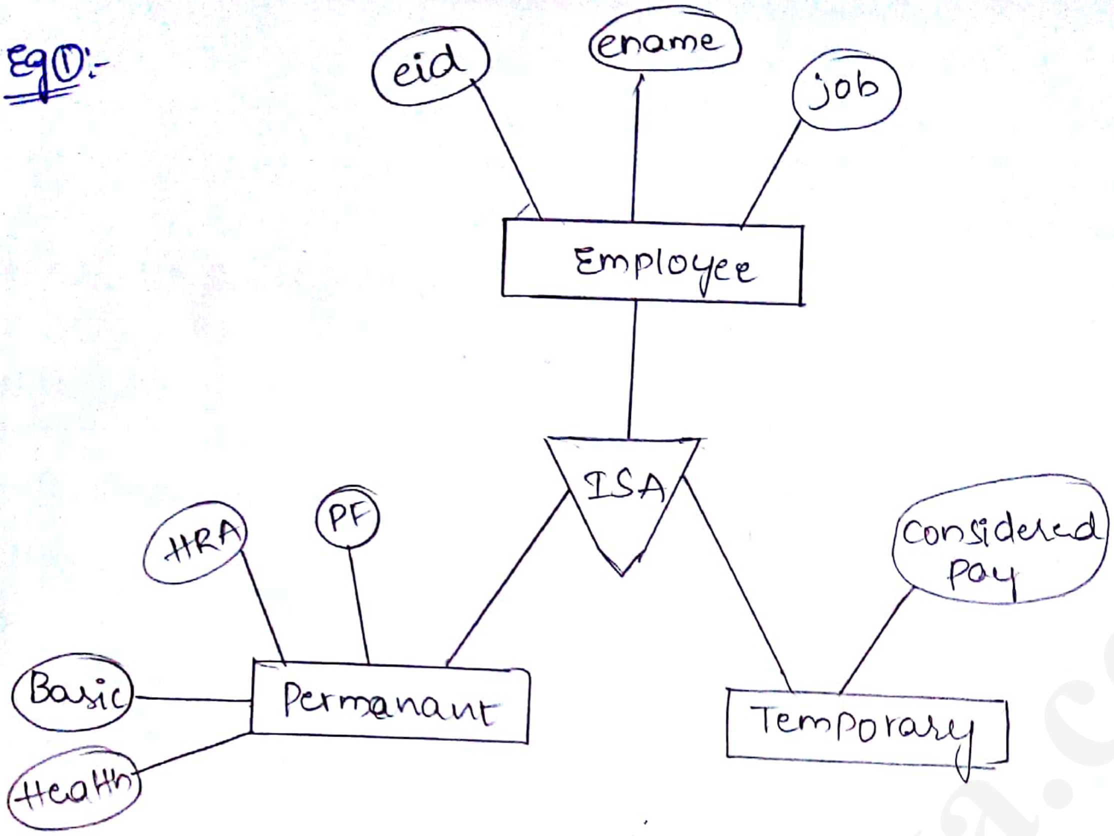
- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

@codes.learning

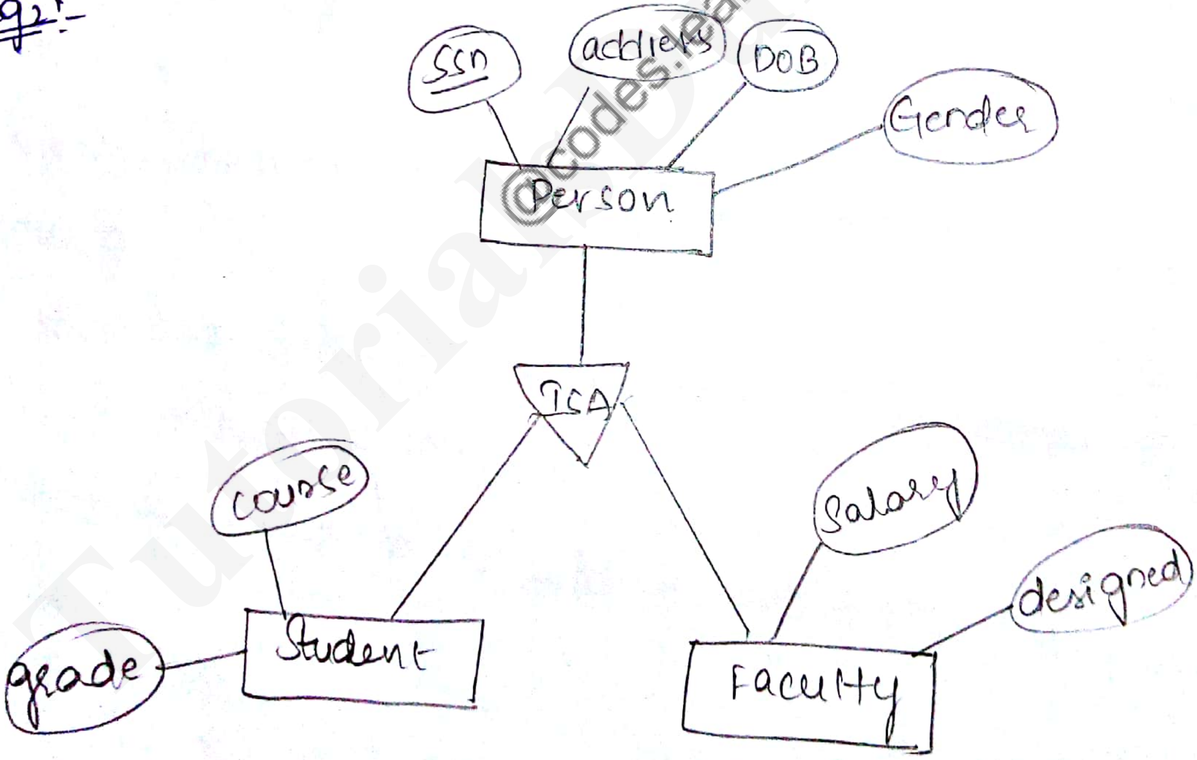
Eg①:-



eid
ename
job
Employee
ISA
HRA
PF
Considered pay
Basic
Permanant
Temporary
Health

→ with covering constraints Permant & Temperory cover Employee.

Eg2:-



ssn
address
DOB
Gender
Person
ISA
course
Salary
designed
grade
Student
Faculty

overlapping constraint.

→ Student overlaps faculty.

Q:What is a view? How views are implemented?

A) **VIEW:-**

A view is nothing more than a SQL statement that is stored in the database with an associated name. A view is actually a composition of a table in the form of a predefined SQL Query.

A view can contain all rows of a table or select rows from a table. A view can be created from one or many tables which depends on the written SQL Query to create a view.

views, which are a type of virtual tables allow users do the following

- structure data in a way that users or classes of user find natural or intuitive
- Restrict access to the data in such a way that a user can see

creating views:-

Database views are created using the CREATE VIEW statment. views can be created from a single table, multiple tables or another view.

TutorialsDuniya.com @codes.learning

To create a view, a user must have the appropriate system Privilege according to the specific implementation

The basic CREATVIEW Syntax as follows

```
CREAT VIEW View-name as
SELECT column1, cloumn2 - - -
FROM table-name
Where [condition];
```

EX:-

consider the CUSTOMERS table having the following records

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	khilan	25	Delhi	1500.00
3	Kaushik	23	Kota	2000.00
4	chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	MUFFY	24	Indore	10000.00

Following is an example to create a view from the ~~cs~~ CUSTOMERS table. view would be used to have customer name and age from the CUSTOMER table.

```
SQL >CREATE VIEW CUSTOMERS_VIEW AS
      SELECT name,age
      FROM CUSTOMERS;
```

Now, You can Query CUSTOMERS_VIEW in a similar way as You queries actual table. Following is an example for the Same.

```
SQL > Select * From CUSTOMERS_VIEW;
```

This would Produce the following result

Name	age
Ramesh	32
Khilan	25
kaushik	23
chaitali	25
Hardik	27
Komal	22
MUFFY	24

Updating a view:-

A view can be Updated under certain conditions which are given below

* The SELECT clause may not contain the keyword DISTINCT

* The SELECT clause may not contain summary functions

* The SELECT clause may not contain set functions
* The SELECT clause may not contain set operators
* The SELECT clause may not contain an ORDER BY clause
* The FROM clause may not contain multiple tables
* The Where clause may not contain subqueries
* The query may not contain GROUP BY or HAVING
* Calculated columns may not be updated

So, if a view satisfies all the above-mentioned rules then you can update the view. The following code block has an example to update the age of Ramesh.

```
SQL >UPDATE CUSTOMERS_VIEW
        SET AGE =35
      Where name = 'Ramesh';
```

ID	Name	Age	ADDRESS	SALARY
1	Ramesh	35	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	Kaushik	23	Kota	2000.00
4	chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Inserting Rows into a view:-

Rows of data can be inserted into a view. The same rules that apply to the UPDATE command also apply to the INSERT command.

Here, we cannot insert rows in the CUSTOMERS_VIEW because we have not included all the NOT NULL columns in this view, otherwise you can insert rows in a view in a similar way as you insert them in a table

Deleting Rows into a view:-

Rows of data can be deleted from a view. The same rules that apply to the UPDATE and INSERT commands Apply to the DELETE command

Following is an example to delete record having AGE=22

```
SQL > DELETE FROM CUSTOMERS_VIEW
          where age = 22;
```

This would ultimately delete a row from the base table CUSTOMERS and the same would reflect in the view itself. Now, try to query the base table and the SELECT statement would produce the following result

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	35	Ahmedabad	2000.00
2	khilan	25	Delhi	1500.00
3	kaushik	23	kota	2000.00
4	chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
7	muffy	24	Indore	10000.00

Dropping views:-

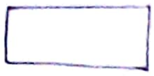
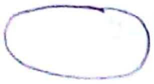
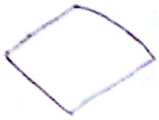
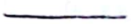
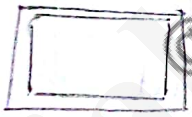
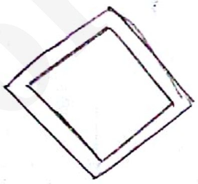
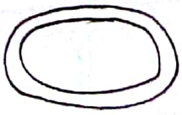
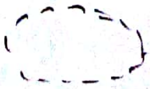
obviously, where you have a view, you need a way to drop the view if is not longer need.

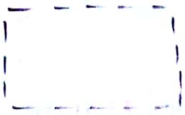
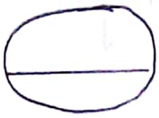
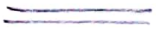
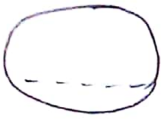
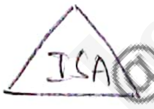
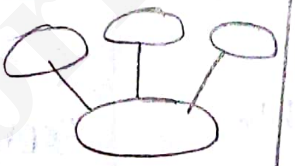
Syntax:-

DROP VIEW view-name;

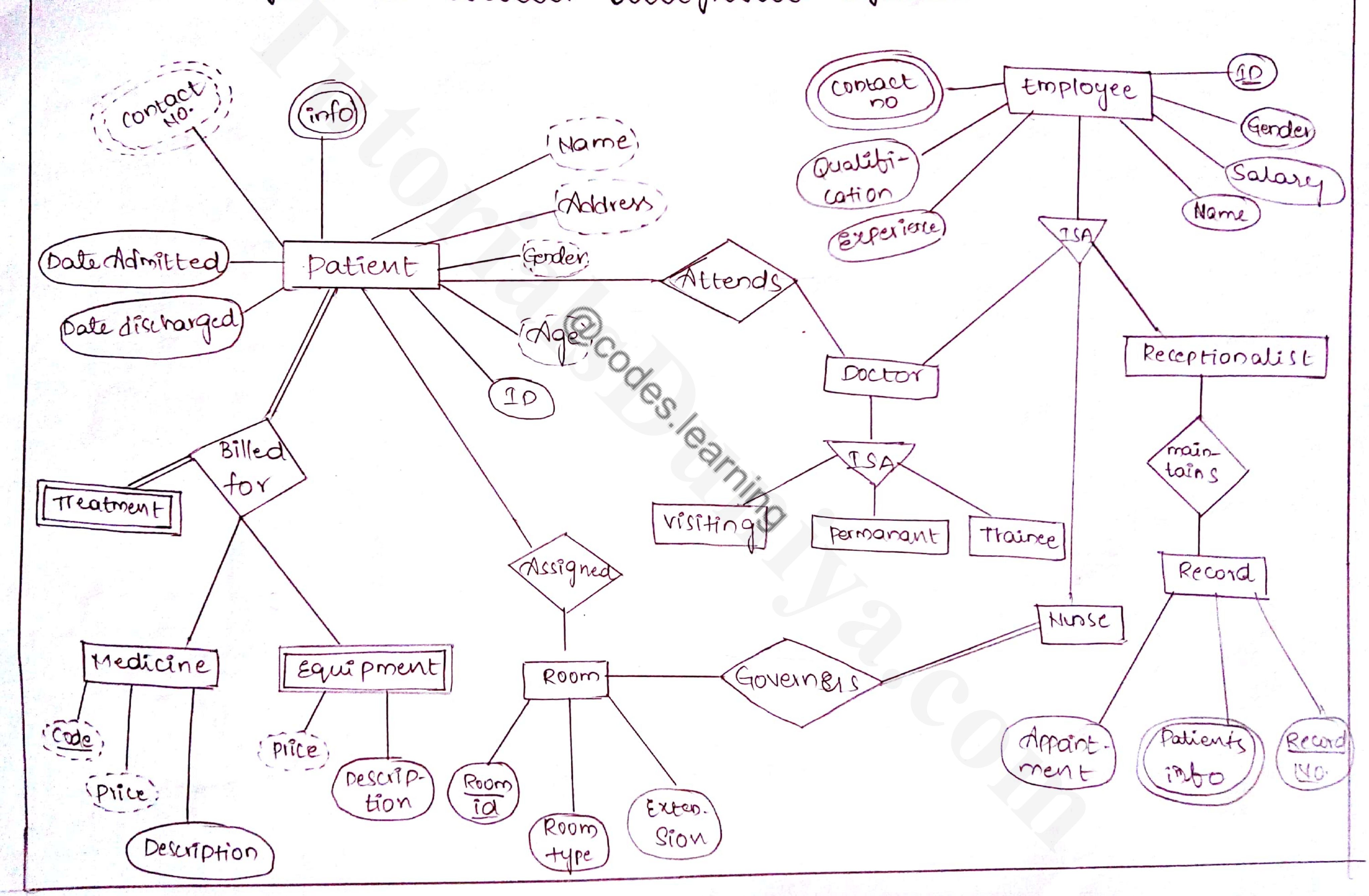
Q.- Draw an ER-diagram for Hospital Management System.

A) Notations used for the elements of ER diagram:-

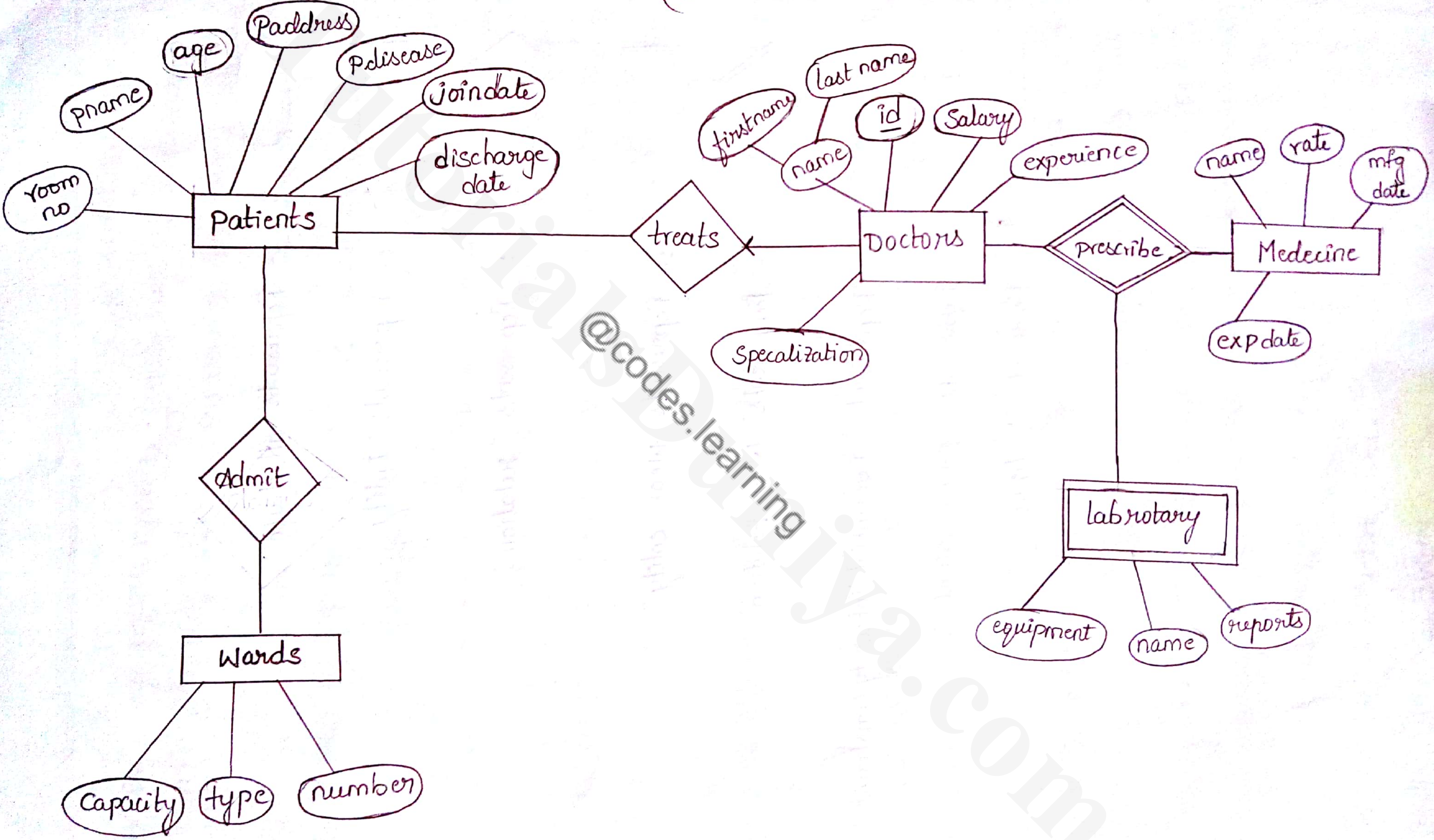
ER Element	Symbol	Symbol Name	Function
Entity		Rectangle	Represent Entity Sets.
Attribute		Ellipse	Represent attributes.
RelationShip		Diamond	Represents relationship Set among Entity Sets.
Link		Line	Link attributes to Entity Sets & Entity Sets to relationship Set.
Weak Entity		Double Rectangle	Represents weak Entity Set
Weak RelationShip Set		Double diamond	Represents relationship Set that is associated with a weak EntitySet.
Multivalued attribute		Double Ellipse	Represents multi-valued attribute.
Derived attribute		Dashed Ellipse	Represents derived attribute.

ER Model	Symbol	Symbol name	Function.
Aggregate Entity Set		Dashed rectangle	Represent aggregate Entity Set
Key attribute		underline Ellipse	Represents primary key attribute.
Mandatory Relationship		Double lines	Represents total participation.
Partial key attribute		Dashed underline Ellipse	Represents partial key attribute.
class hierarchy		Triangle labeled ISA	Represents generalization Specialization.
composite attribute			Represent Composite Attribute.

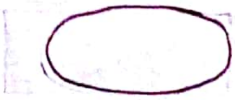
* E-R Diagram for Hospital Management Systems *


contact NO.
info
Name
Address
Gender
Date Admitted
Patient
Date discharged
Age
ID
Attends
Billed for
Treatment
Medicine
Code
Price
Description
Equipment
Price
Descrip-tion
Assigned
Room
Room id
Room type
Exten. Sion
Governs
Contact no
Employee
ID
Gender
Qualifi-cation
Salary
Name
experience
ISA
Doctor
ISA
Visiting
Permanant
Trainee
Receptionalist
main-tains
Record
Nurse
Appoint-ment
Patients info
Record NO.

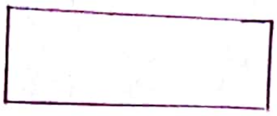
(Or)



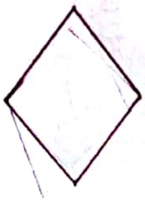
Patients
room no
pname
age
Paddress
Pdisease
joindate
discharge date
treats
Doctors
firstname
name
last name
id
Salary
experience
Specalization
Prescribe
Medecine
name
rate
mfg date
expdate
labrotary
equipment
name
reports
Admit
Wards
Capacity
type
number



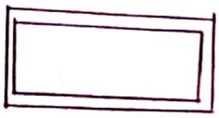
:- Represents "attribute"



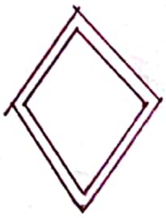
:- Represents "Entity"



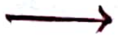
:- Represents "relation"



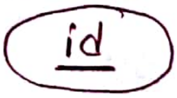
:- Represents "Weak entity"



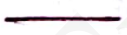
:- Represents "Weak relation"



:- Represents "Referential Integrity Constraint"



:- Represents "prime attribute".



:- Represents "links".

1. Explain E-R MODEL With a Suitable Example.

Ans: Entity Relationship Model:

* A popular High level conceptual object based data model proposed by "peter in 1976."
* It allows to describe the data involved in a real world interms of entities or objects and their relationships.
* It is graphically represented in the form of E-R diagrams.
* key elements of ER model are entries, attributes and relationships.

Advantages:

⇒ the constructs used in the ER model can easily be transformed into relational tables.

⇒ It is independent to the specific DBMS.

⇒ It is simple and easy to understand.

⇒ The db designer can use this model to communicate the design to the end user.

⇒ the model can be used as a design plan by the db developer to implement a data model in specific db management s/w.

tutorialsduniya.com @codeslearning

Basic Concepts of ER model:-

1) a) Entity: An entity is any distinguishable object or thing that exist in the real world (person, place, object, event or concept).

→ A distinguishable object about which data needs to be collected.

→ An entity may be an object with a physical existence like particular person, car, house or employee.

1) b) entity set or Entity class:-

collection of similar entities. A set of all entities of same type is called an entity set.

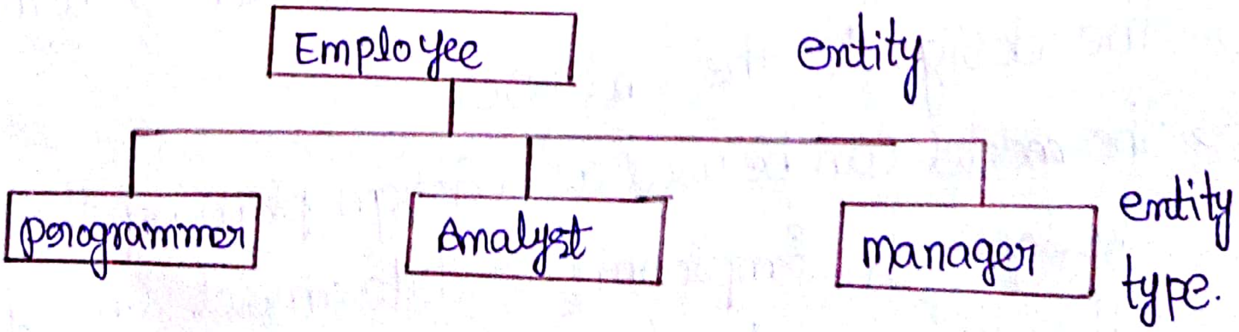
→ An entity set is represented as Rectangle in ER model

NOTE:

- An entity set is analogous to a table in the Relational model
- An entity is analogues to a row in the relational model.

1) c) Entity Type:

* Sub type of given entity, which is again consider-ed as seperate entities is called entity type



Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

@codes.learning

2) Attributes :-

- A property or characteristic of an entity that is used to describe the entity
- An entity is represented by a set of attributes. i.e attributes describe the entity of which they are associated.

EX: A consumer entity has the attributes names (actual) customer number, name, Date of birth, address, phoneno.

NOTE: An attribute is analogous to a column in the relational model · Entity set = table · Entity-name

TYPES OF ATTRIBUTES:

Attributes can be

i) mandatory or optional:

Mandatory attribute is one for which a value must be known

- optional attribute is one for which a value may not necessarily be known

EX: customer-number attribute of a customer entity set is manditory. But phno attribute of a customer entity set can be optional.

ii) Stored or derived:

- An attribute whose values are stored in the db is called stored attribute.

https://www.tutorialsduniya.com Scanned by CamScanner

- An attribute 'experience' of an employee is computed from joining date.

→ Derived attributes need not be stored in the physical tables.

iii) Single valued or multivalued :-

- An attribute that can take only one value for an entity is known as single valued attribute.

one customer number, but he can have two addresses say office address and Residential address.

iv) Simple or composite :-

An attribute that cannot be broken down into smaller components. Ex: color, age, weight

An attribute that can be broken down into component parts is known as composite attributes.

Ex: A consumer name can be further classified into first name, middle name & last name

v) Key or non-key :-

An attribute or minimal set of attributes whose value uniquely identifies an entity in the entity set is called key attribute. Ex: 'Rno' is a key attribute of student entity set as 'Rno' installed be unique, 'name' is a non key attribute.

@cbdeslearning

(3) Relationships and Relationshipset :-

⇒ Association among two or more entities is called relationship i.e A relationship represent an association b/n two or more entities.

Eg: employee & Department participate in the relationship 'works-In'

· customer & Account participate in the relationship 'deposito

A relationship set has its own attributes.

- Several relationship sets may involve in the same entity sets.

4) E-R Diagram:

E-R model is graphically represented in the form of ER diagrams.

E-R diagram contains 3 main elements.

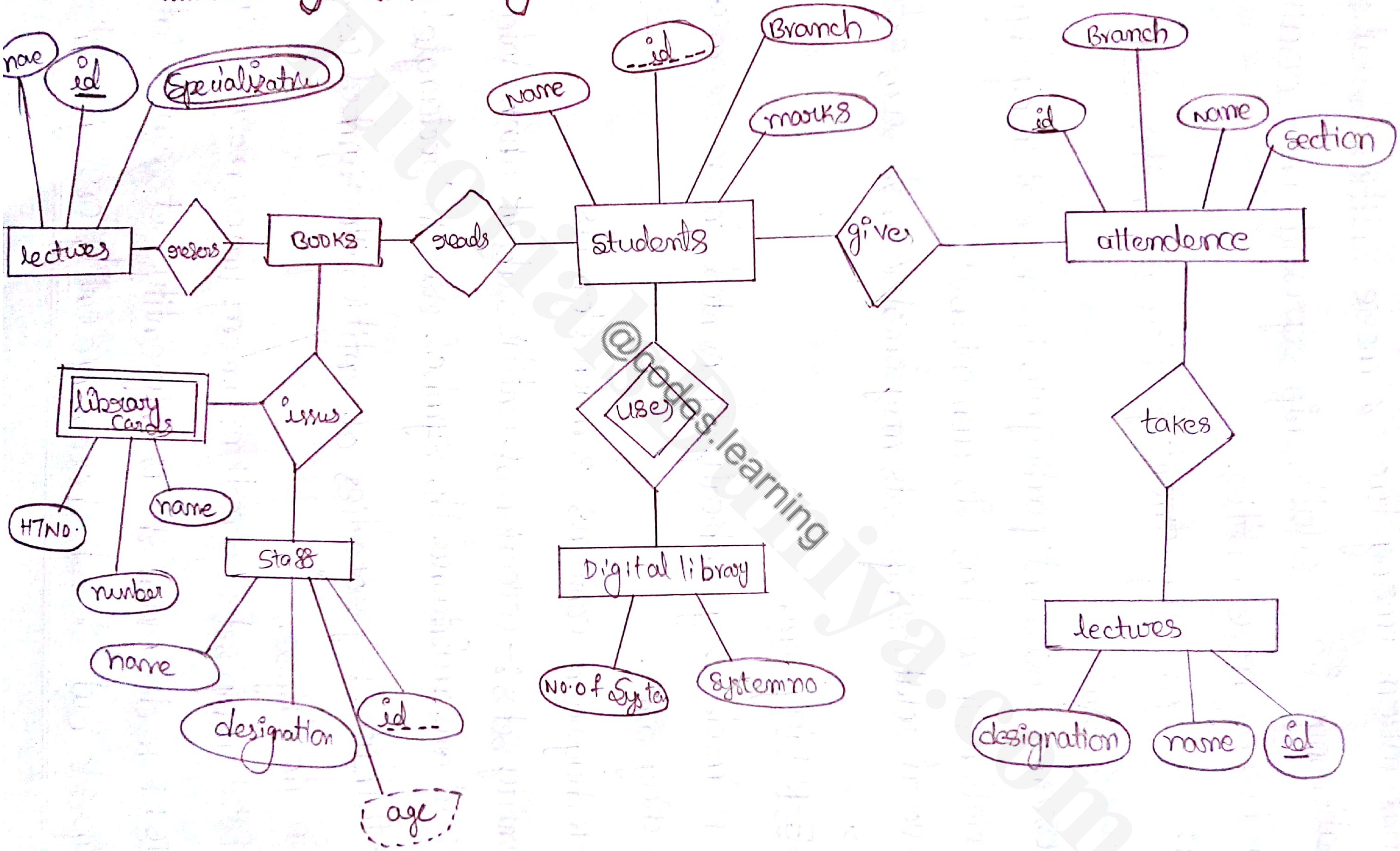
i) entity sets :- Collection of entities about which, data is stored and maintained the symbol used in 'rectangle'.

ii) Attributes :- properties of entities. The symbol used in the ellipse.

iii) Relationship set : connections among two or more entities

NOTE: E-R diagram shows only entity sets and relatio -shipset but do not show single entity or single relationshi

Library Management Systems


nae
id
Specialization
lectures
refers
Books
reads
Name
id
Branch
marks
students
give
id
Branch
name
section
attendence
Library Cards
issus
uses
takes
H7NO.
name
number
Staff
name
designation
id
age
Digital library
No.of Sys.ter
Systemno
lectures
designation
name
id

* What do you understand by the mapping cardinalities?

Ⓐ

Cardinality of relationship (or) Mapping cardinalities·

→ The cardinality of a relationship or connectivity of a relationship descibre the mapping of associated entity in the relationship set·

→ Mapping cardinalities express the no·of entities to which another entity can be associated via a relationship

→ Mapping cardinalities are most useful in describing binary relationship sets, the value of mapping cardinalites are one or many·

→ The basic types of connectivity for relation are

1· one - to - one (1:1)

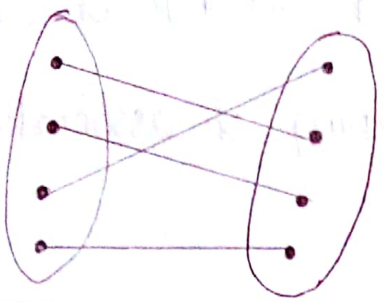
2· one - to - many (1:M)

3· Many - to - one (M:1)

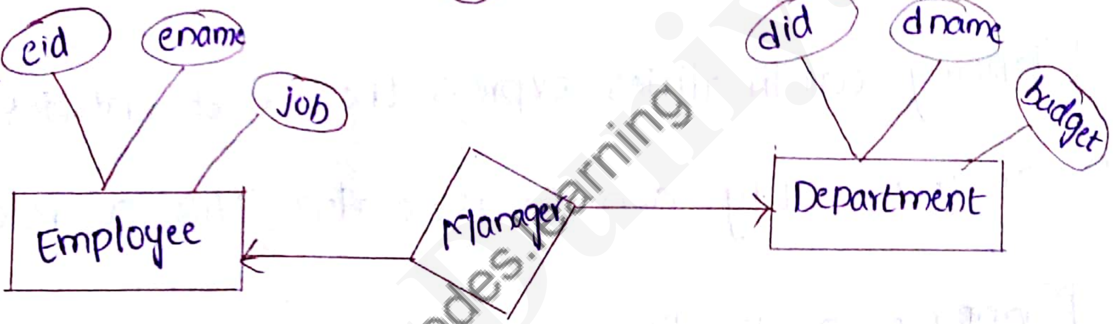
4: Many to Many (M:M)

1. One-to-one :-

An entity in 'A' is associated with at most one entity in B, and an entity in B is associated with at most one entity in 'A'.

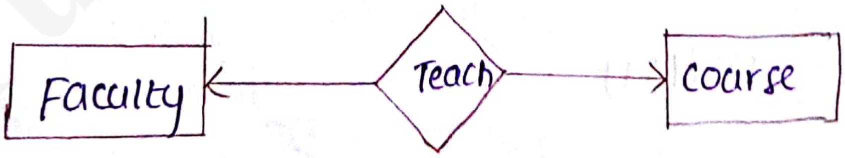


Ex ①:-


eid
ename
job
Employee
Manager
did
dname
budget
Department

- An Employee can manage only one Department (Act as manager, or HOD)
- A Department can be managed by only one Person

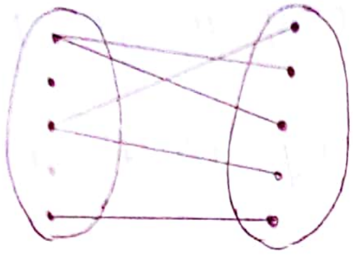
Ex ②:-


Faculty
Teach
Course

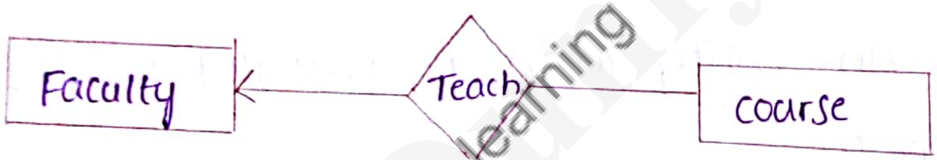
- Each Faculty member will teach no more than one course, and each course will be taught by no more than one faulty members.

2. one - to - many (1:N)

An entity in 'A' is associated with any no. of entities (zero or many) entities in 'B' but an entity in B can be associated with atmost one entity in A.



Ex:

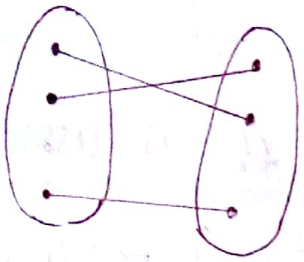


- one faculty can teach many courses like c, c++
- A course can be handled by only

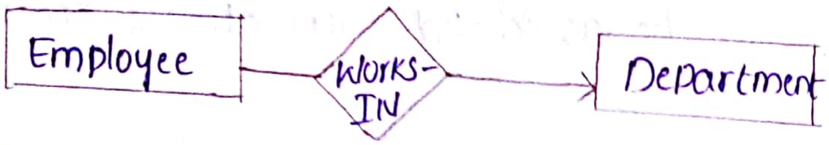
3. Many-to-one: (M:1)

An entity in 'A' is associated with at most one entity in 'B'.

An entity in 'B' is associated with many entities in 'A'.



Ex:-


Employee
Works-IN
Department

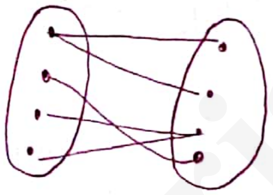
* An Employee can work only in one Department.

* Department can have any no. of employees.

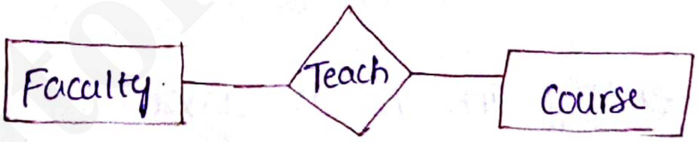
4. Many-to-Many (M:N):-

→ An entity in 'A' is associated with no. of entities in B.

An entity in 'B' is associated with any no. of entities in 'A'.



Ex:


Faculty
Teach
Course

* Each Faculty member may teach several courses.

* A course can be taught by several Faculty members

@codesislearning

⑧ Explain various about various Constraints used in E-R model)

- Constraints are used to limit the type of data that can go into a table.
- constraints can be specified when a table is created (with CREATE TABLE statement) or after the table is created (with the ALTER TABLE STATEMENT)

We will focus on the following constraints:

NOT NULL
UNIQUE
PRIMARY KEY
FOREIGN KEY
CHECK
DEFAULT

---

* NOT NULL Constraint

- By default a table column can hold NULL values.
- The NOT NULL constraint enforces a column to NOT accept NULL values.
- The NOT NULL constraint enforces a field to always contain a value. This means that you cannot insert a new record, or update a record without adding a value to this field.

- The following SQL enforces the "P_Id" column and the "LastName" column to not accept NULL values:

```
↳ Create table persons
  (
    P_Id int NOT NULL,
    LastName varchar(25) NOT NULL,
    FirstName varchar(25),
    Address varchar(25),
    City varchar(25)
  );
```

---

* UNIQUE Constraint

- The UNIQUE constraint uniquely identifies each record in a database table.
- The UNIQUE and PRIMARY KEY constraints both provide a guarantee for uniqueness for a column or set of columns.
- A PRIMARY KEY constraint automatically has a UNIQUE constraint defined on it.
- Note that you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

- **UNIQUE Constraint on CREATE Table:**

The following SQL creates a UNIQUE constraint on the "P_Id" column when the "persons" table is created.

* PRIMARY KEY Constraint

- The PRIMARY KEY constraint uniquely identifies each record in a database table.
  - Primary keys must contain unique values
  - A primary key column cannot contain NULL and Duplicate values.
  - Each table should have a primary key, and each table can have only one primary key.

- PRIMARY KEY Constraint on CREATE TABLE
  - The following SQL creates a PRIMARY KEY on the "P_Id" column when the "persons" table is created.

↳
```
Create table persons
 (
   P_Id int Not NULL,
   LastName varchar(25) NOT NULL,
   FirstName varchar(25),
   Address varchar(25),
   City varchar(25),
   PRIMARY KEY(P_Id)
 )
```

- TO DROP a PRIMARY KEY Constraint

MYSQL:
```
ALTER table persons
DROP PRIMARY KEY
```

Tutorials Duniya.com @codes.learning

SQL Server:

```
Alter table persons
Drop Constraint PK_personID
```

---

* FOREIGN KEY Constraint

- A FOREIGN KEY in one table points to a PRIMARY KEY in another table.
- Let's illustrate the foreign key with an Example. Look at the following two tables

The person's table

P_Id	LastName	FIRSTNAME	Address	City
1	FIROZ	Pranav	Vpura	Gunteur
2	AJAY	Raghu	Yelhanka	Banglore
3	LAKSHMAN	Shravan	Hebbal	Hydrabad

The order's table

O_Id	OrderNo	P_Id
1	77895	3
2	44678	3
3	22456	2
4	52211	1

- Note that the "P_Id" column in the "orders" table points to the "P_Id" column in the "persons" table.
- The "P_Id column in the "persons" table is the PRIMARY KEY in the "persons" table
- The "P_Id" column in the "orders" table is a FOREIGN KEY in the "orders" table.

Ex:

```
Create table persons
  (
   P_Id int NOT NULL, LastName varchar(25) NOT NULL,
   FirstName varchar(25), Address varchar(25),
   City varchar(25), CHECK(P_Id >0)
  )
```

---

* DEFAULT Constraint

- The DEFAULT constraint is used to insert a default value into a column.
- The default value will be added to all new records, if no other value is specified.

Ex: . The following SQL creates a Default constraint on the "city" column when the "persons" table is created.

```
↳ Create table persons
    (
      P_Id int NOT NULL,
      LastName varchar(25) NOT NULL,
      FirstName varchar(25),
      Address varchar(25).
      City varchar(25) DEFAULT 'sandnes'
    )
```

- The FOREIGN KEY constraint is used to prevent actions that would destroy link between tables.

- The FOREIGN KEY constraint also prevents that invalid data is inserted into the foreign key column because it has to be one of the values contained in the table it points to.

↳ creating table.

```
Create table orders
    (
      O_Id int NOT NULL, orderNo int NOT NULL,
      P_Id int, Primary Key(O_Id),
    , FOREIGN KEY (P_Id) REFERENCES persons(P_Id)
    )
```

- DROPing a FOREIGN KEY

```
Alter table orders
DROP FOREIGN KEY fk_perOders
```

- SQL server

```
Alter table Orders
Drop Constraint fk_perOrders
```

★ CHECK Constraint

- The CHECK constraint is used to limit the value range that can be placed in a column.

- If you define a CHECK constraint on a single column it allows only certain values for this column.

- If you define a CHECK constraint on a table it can limit the values in other columns in the row.

15F01A05E9 Karim Basha

1) Differentiate between independent and Correlated nested Queries?

Ans There are mainly two Types of nested Queries.

They are: 1. independent nested Queries
2. Correlated nested Queries.

1) Independent nested Queries:-

In independent nested Queries, query execution starts from innermost query to outermost queries. The execution of inner query is independent of outer query, but the result of inner query is used in execution of outer query. Various operators like IN, NOT IN, ANY, ALL etc are used in writing independent nested queries. It is also called as incorrelated subquery.

Example:

```
SELECT salesperson. Name from salesperson WHERE
salesperson.ID NOT IN (SELECT orders. salesperson_id from
orders, customers WHERE orders. cust_id = customer.ID AND
customer.Name = 'samsonic');
```

2) correlated nested queries:-

A correlated subquery is one that is executed after The outer query is execated. So correlated subqueries take an approch opposite to that of normal queries. It is also called as dependent nested query. It can be executed for each row of the result.

Example:- Select* from employee;

```
SELECT employee_number, name
FROM employees As emp
WHERE salary > (
SELECT AVG (salary)
  FROM employees
WHERE department =emp. department);
```

in the above query the outer query is:

```
SELECT employee_number, name
 FROM employee As emp
 WHERE salary > ----
```

and the inner query is:

```
SELECT AVG (salary)
FROM employees
WHERE department = emp. department.
```

In above nested query the inner query has to be re-executed for each employee.

SK. MUBEENA
15F01A05F0

5a) Consider the following schemas:

Sailors (sid, sname, rating, age)

Reservers (sid, bid, day)

Boats (bid, bname, color)

Write the following queries in relational algebra, tuple relational calculus and domain relational calculus:

a) Find the name of sailors who have reserved boat 103.

b) Find the names of ages of sailors with a rating above 7

c) Find the names of sailors who have reserved a read boat.

d) Find the sname, bid, and day for each reservation

e) Find the name of sailors who have reserved atleast one boat.

b) Draw a ER diagram for Hospital Management System

5a)

a) Find the name of sailors who have reserved boat 103

Query:-

```
Select  S.sname  from  sailors S , Reservers R  .
                 where  S.sid = R.sid  and  R.bid = 103;
```

b) Find the names and ages of sailors with a rating above 7 ? .

Query :-

```
Select    sname , age
from      sailors
Where     rating >7   ;
```

c) Find the names of sailors who have reserved a red boat .

Query:-

```
select   S.sname
from     Sailors S , Reservers R, Boats B
Where    R.bid = B.bid  and  B.color = red;
```

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

facebook WhatsApp twitter Telegram

d) Find the sname, bid and day for each reservation

Query:-

```
select  s.sname, r.bid, r.day
from    Sailors s, Reservers r
where   s.sid = r.sid;
```

e) Find the name of sailors who have reserved atleast one boat.

Query:-

```
select  s.sname
from    sailors s, Reservers r
where   s.sid = r.sid;
```